Spring 2014

Delay Flip-Flop (DFF) Metastability Impact on Clock and Data Recovery (CDR) and Phase-Locked Loop (PLL) Circuits

Alfred Sargezisardrud
San Jose State University

DELAY FLIP-FLOP (DFF) METASTABILITY IMPACT ON CLOCK AND DATA RECOVERY (CDR) AND PHASE-LOCKED LOOP (PLL) CIRCUITS

A Thesis

Presented to

The Faculty of the Department of Electrical Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Alfred Sargezisardrud

May 2014

The Designated Thesis Committee Approves the Thesis Titled

DELAY FLIP-FLOP (DFF) METASTABILITY IMPACT ON CLOCK AND DATA RECOVERY (CDR) AND PHASE-LOCKED LOOP (PLL) CIRCUITS

By

Alfred Sargezisardrud

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2014

Dr. Shahab Ardalan	Department of Electrical Engineering
Dr. Sotoudeh Hamedi-Hagh	Department of Electrical Engineering
Dr. M. J. Zoroofchi	Department of Electrical Engineering

ABSTRACT

DELAY FLIP-FLOP (DFF) METASTABILITY IMPACT ON CLOCK AND DATA RECOVERY (CDR) AND PHASE-LOCKED LOOP (PLL) CIRCUITS

by Alfred Sargezisardrud

Modeling delay flip-flops for binary (e.g., Alexander) phase detectors requires paying close attention to three important timing parameters: setup time, hold time, and clock edge-to-output (or briefly C2Q time). These parameters have a critical role in determining the status of the system on the circuit level. This study provided a guideline for designing an optimum DFF for an Alexander phase detector in a clock and data recovery circuit. Furthermore, it indicated DFF timing requirements for a high-speed phase detector in a clock and data recovery circuit. The CDR was also modeled by Verilog-A, and the results were compared with Simulink model achievements. Eventually designed in 45 nm CMOS technology, for 10 Gbps random sequence, the recovered clock contained 0.136 UI and 0.15 UI peak-to-peak jitter on the falling and rising edges respectively, and the lock time was 125 ns. The overall power dissipation was 21 mW from a 1 V supply voltage. Future work includes layout design and manufacturing of the proposed design.

ACKNOWLEDGEMENTS

I am extremely grateful to my advisor, Dr. Shahab Ardalan, for his invaluable guidance and support from the point I started working on the thesis up until this point that I'm presenting the results. Analog mixed signal (AMS) design has been more meaningful for me since I started cooperating with you on various projects. Thanks for providing an exceptional lab with the most up-to-date simulation software tools. Without your help, acceptable results would not be achieved.

I am also thankful to other members of the thesis committee, Dr. Sotoudeh Hamedi- Hagh and Dr. Mohammad Javad Zoroofchi. Your outstanding lectures helped me out in better understanding of analog and digital design methodologies.

Eventually I want to express my deepest gratitude to my family for their magnificent physical and spiritual support. Indeed, without your sincere guidance, I would not be at this point. I consider myself a very lucky person to have you beside me at all times. You are and will be in my heart forever.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

s	Second
ns	Nanosecond
ps	Picosecond
V	Volt
mV	Millivolt
PP	Peak-to-peak
mW	Milliwatt
UI	Unit interval
Gbps	Gigabit per second
μA	Microampere
KHz	Kilohertz
MHz	Megahertz
GHz	Gigahertz
DSM	Delta-sigma modulator
KΩ	Kiloohm
nF	Nanofarad
pF	Picofarad
CMOS	Complementary metal oxide semiconductor
NMOS	N-type metal oxide semiconductor
PMOS	P-type metal oxide semiconductor
DC	Direct current
deg	Degree

Chapter 1. Introduction

1.1. DFF and Transistor Timing Metrics

A delay flip-flop samples the data on the rising (or falling) edge of the clock, and the result is a delayed replica of the input. When a data stream is sampled by the rising (or falling) edge of the clock, the information remains unchanged until the next sampling edge of the clock. This means the information is held independently until the next sampling edge of the clock. Note that the delays caused by the parasitics are neglected. To implement the system on the circuit level, three important timing parameters have to be considered. First, the time that the data become valid before the clock transition is called "setup time." Second, the time that the data become valid after the clock transition is called "hold time." Third, the time elapsed between the clock transition point and the point at which data become stable is called "clock edge-to-output delay" or "C2Q time." This time (C2Q) is always greater than or equal to hold time. Furthermore, C2Q time has to be greater than zero. With a zero delay from the clock port to the output, a DFF could be replaced with a wire. Note that setup and hold times could be zero. These parameters determine the delay of the circuit and also identify the speed of DFFs [1]. Overall, for perfect functionality, these timing parameters have to be adjusted carefully. Figure 1.1 depicts setup, hold, and C2Q times qualitatively.

If setup and hold time requirements are not fulfilled completely, the sampling will not be performed correctly, leading to incorrect results at the output. The reason is that

there would not be enough time for input data to toggle between the signal levels (i.e., high and low levels). Eventually the data will remain idle because of setup or hold time violation(s) and sustain the instant value, obtained from the previous successful sampling action. This malfunction leads to a drawback in reliability (i.e., the system's ability to perform properly for a particular period of time) [2].

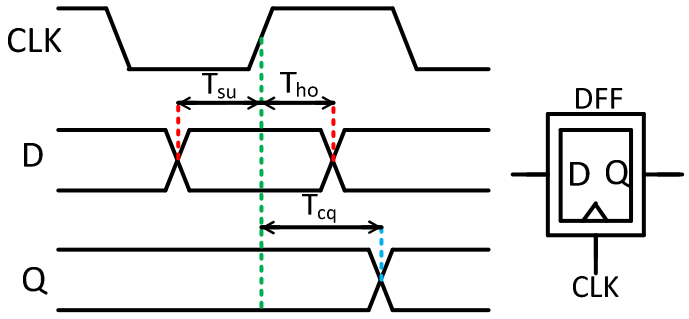


Figure 1.1: DFF, setup, hold, and C2Q times

Due to phase differences between data and clock signals, maximum allowable setup and hold times are determined. According to Figure 1.2, if the clock transition occurs exactly in the middle of the data eye, the maximum allowable setup and hold times will be equal. Any other phase difference will result in unequal maximum allowable setup and hold times.

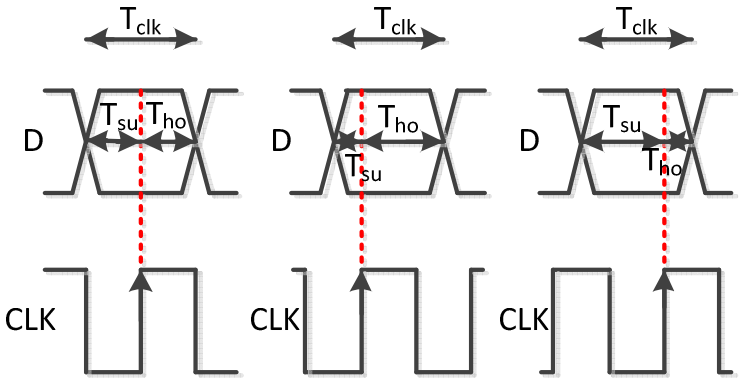


Figure 1.2: Maximum allowable setup and hold times

Assume that the input data have the same value for two clock periods. As depicted in Figure 1.3, if the rising edges of the clock sample the points where data make a transition, then because of setup and hold time violations, the results at the output would be totally different from the predicted ones. However, sampling the data at nontransition points gives us the correct results. Therefore, the sampling point has a critical role in the system performance [1].

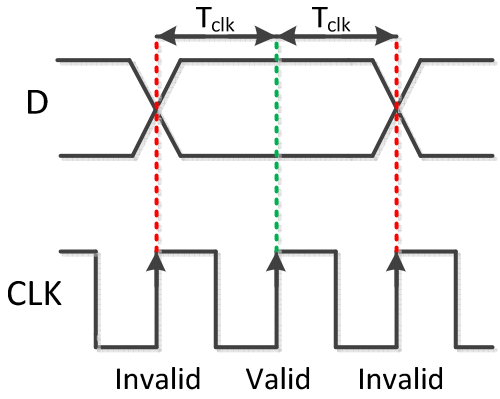


Figure 1.3: Valid and invalid sampling according to clock edge position

If a couple of DFFs are present in the system, each of them has to be optimized according to a specific location in the circuit (i.e., each DFF has to be allocated certain setup, hold, and C2Q times). According to Figure 1.2, if the clock samples the data exactly in the middle of data eye, then the setup and hold times would have the greatest margins, and eventually skew and jitter tolerance would be maximum. As demonstrated in Figure 1.4, FF1 and FF3 are sampling the data on the different edges of the clock. While the rising edges of FF3 sample the data at a point where no data transition occurs, FF1 samples the data exactly at data transition points. Obviously, setup and hold time

requirements have not been met at data transition points. It means setup and hold violations are observed at those points. To resolve this problem, the data and clock phase difference has to be changed [1].

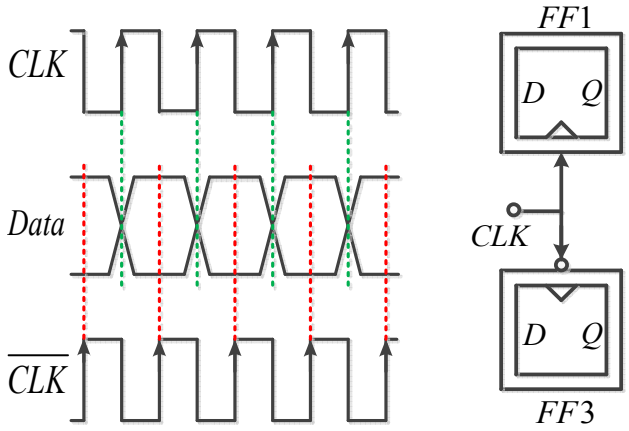


Figure 1.4: Sampling points for FF1 and FF3

1.2. PLL and CDR

Due to the exponential growth of internet node numbers, the amount of data transported rises quickly. To support the high volume of data transmission, naturally high bandwidth is required [3]. To transfer the data, cables are utilized. However, the use of cables for high-speed applications is limited due to low bandwidth and high information loss. The alternative solution for data transmission is optical fiber. The data at the transmitter side are converted to light by laser diodes and then transmitted by optical fibers. At the receiver side, photo diodes convert the information to current. Optical fibers, compared with cables, have higher bandwidth and lower data loss. Receivers incorporate CDRs to extract clean clock and data from the noisy information

delivered by optical fibers [2].

Figure 1.5 depicts the extraction process. The decision circuit block retimes the input data, and the result is a delayed input signal with less noise on it. The extraction quality is extremely important for the next stages. The generated clock at the output of the clock recovery circuit must include three specific features. First, its frequency must be equal to the input data rate. Second, it must have a phase difference with respect to the input data. Third, it must carry small noise [4].

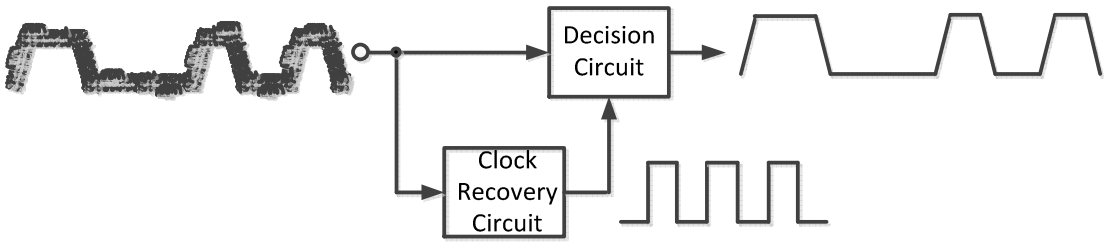


Figure 1.5: Clean data and clock extraction from noisy input data

1.3. PLL and CDR Market Overview and Applications

With an increasing number of YouTube®, *Facebook*®, and smart phone consumers, the need for higher bandwidth and stronger data processing devices is inevitable. Cisco, Inc. predicted that global internet traffic would surpass one zetta-byte in 2016 [5]. Figure 1.6 shows the increasing internet traffic over time according to the Cisco connected world technology report [5]. Note that the internet traffic in 2016 is expected to be five times higher than the amount in 2011. To afford the high volumes,

SerDes (i.e., Serializer-Deserializer) circuits must satisfy tighter performance specifications along with lower bit error rates (BERs). Extremely low BERs in the range of 10^{-12} to 10^{-15} urge analog mixed-signal designers to implement highly efficient PLLs and CDRs with extremely low jitter components. Figure 1.7 illustrates the exponentially growing link speeds over time for various I/O (i.e., input/output) standards. With growing bit rates for each I/O standard, CDR design specifications become more demanding and complicated [6].

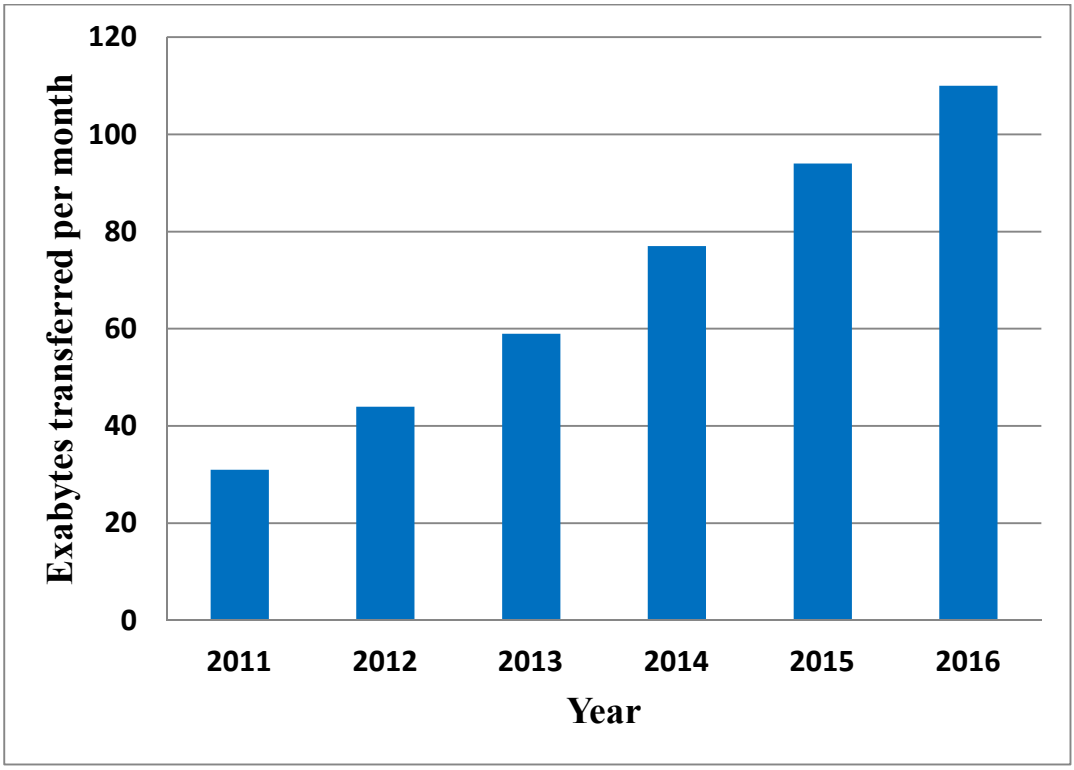


Figure 1.6: Global internet traffic over time [5]

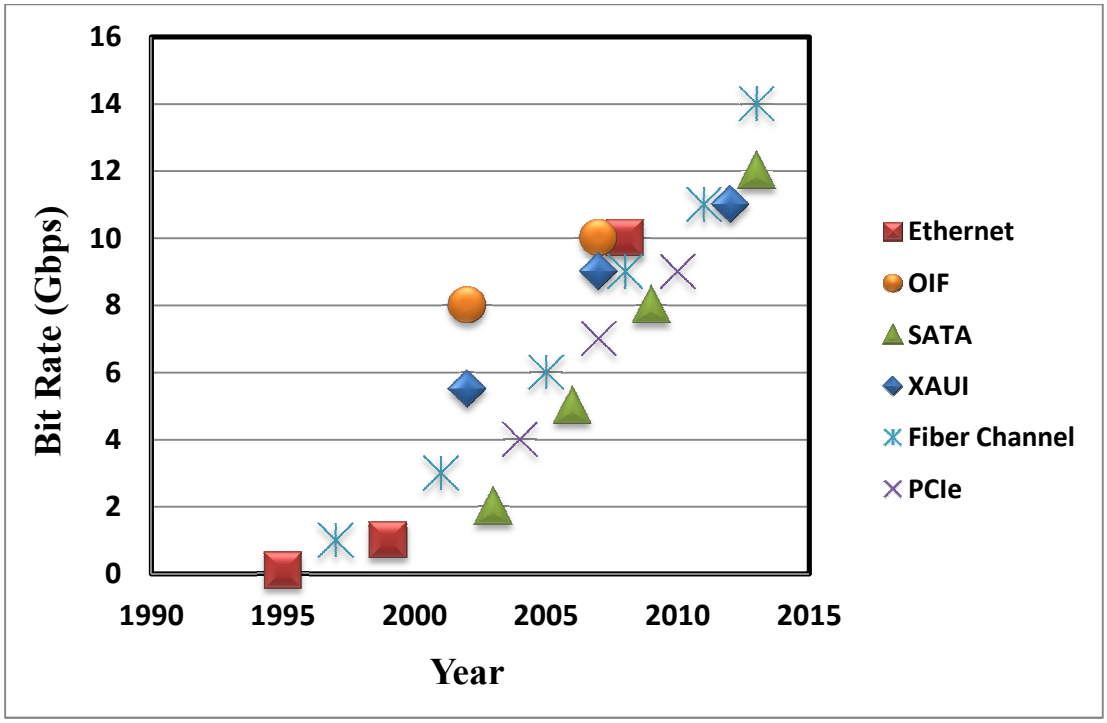


Figure 1.7: Link speed over time for I/O standards [6]

Table 1 summarizes the areas where CDRs and PLLs are being utilized [2]. Clock and data recovery have become the main focus of SerDes designers in industry and academic areas. Each year, sophisticated designs, which overcome some of the challenges, are being proposed in conferences such as the International Solid-State Circuits Conference (ISSCC) and International Symposium on Circuits and Systems (ISCAS). However, a vast region of interest in circuit designs remains untouched and has to be pointed out by designers of highly efficient PLL and CDR circuits. Challenges such as power supply downscaling and power dissipation make clock and data recovery tasks highly demanding. The trade-off between power supply reduction and transistor threshold voltage prevents transistor fabricators from further device size reduction. With

a shrinking size of transistors, power supply decreases, affecting the amount of jitter on the recovered clock and data significantly. Recent designs, performed by Intel and Broadcom, target link speeds of 100 Gbps [6].

Table 1. Areas of interest for CDRs and PLLs

Application	Description
Analog Circuits	1. Reducing jitter on data and clock significantly 2. Providing highly-efficient tunable filter
Digital Circuits	1. Suppressing PLL or CDR phase noise 2. Decreasing systematic skews
Communication Circuits	Recovering data and clock for transmitters and receivers from the signals affected by noise
Radio Frequency Circuits	1. Producing carrier 2. Utilized in modulator or demodulator blocks
Frequency Synthesis	Providing required frequencies wherever frequency multiplication or division is needed

1.4. Motivation and Agenda

The goal of this study was to highlight the impact of metastability on CDR and PLL circuits and to improve the recent CDR designs from noise and lock time aspects. The transistor timing metrics influence DFF outputs, and because DFFs are the main components of CDR circuits, a robust design can be achieved if timing parameters are taken care of precisely. The main task was to build an exceptional CDR from precision and robustness aspects and to achieve optimum results.

Chapter 2 discusses the elements operating inside CDR and PLL circuits. Different types of PLLs were studied and simulated. Jitter and its impact on clock and data recovery circuits were inspected. At the end of this chapter, the areas where CDRs are being utilized are introduced. Chapter 3 presents a novel approach for modeling DFFs and CDRs using transistor timing metrics in Matlab® and Simulink® software. Moreover, a DFF calibration technique was proposed and simulated for different cases. Eventually various phase detector architectures were modeled and applied to the modeled CDR. Chapter 4 describes the CDR implementation in Cadence Virtuoso® software using Verilog-A codes and transistor-level circuits. Every step taken in Chapter 3 was inspected in a different environment.

Chapter 2. Background

2.1. CDR Building Blocks

CDR and PLL circuits incorporate phase detector (PD), charge pump (CP), loop filter (LF), and voltage-controlled oscillator (VCO) to extract clean data and clock signals [2]. CDR architecture is shown in Figure 2.1. The following subsections will explain each block in detail.

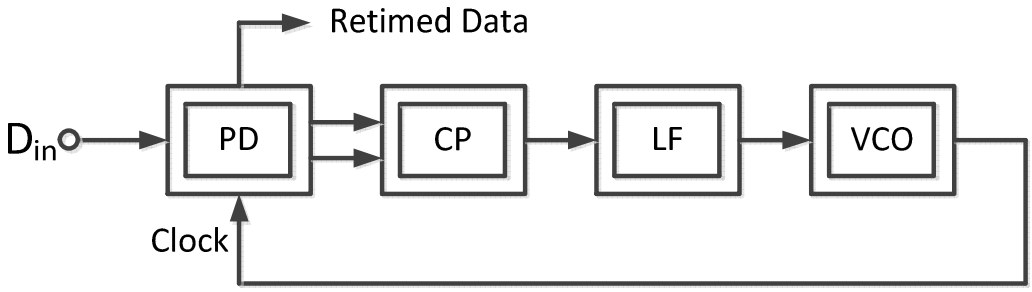


Figure 2.1: CDR building blocks

2.1.1. Phase Detector (PD)

PD is a circuit that compares the phases of its two inputs. This block is considered the brain of PLL and CDR circuits. It decides how the feedback system should react to the clock edge variations. Linear and binary (bang-bang) PDs are utilized in various CDR or PLL architectures. In linear PDs, the output is varied linearly with respect to the phase difference between its two inputs. Figure 2.2 illustrates the characteristic of a linear PD. The slope of the line is called “PD gain” and calculated by the following equation:

$$K_{PD} = \frac{\overline{V_{PD}}}{\Delta\Phi} \tag{2.1}$$

Where K_{PD} is phase detector gain. Note that with increasing phase difference, PD gain will not be reduced. The reason is that when the phase difference between two inputs increases, the average PD output, which is proportional to the phase difference, increases, making the PD gain constant [2].

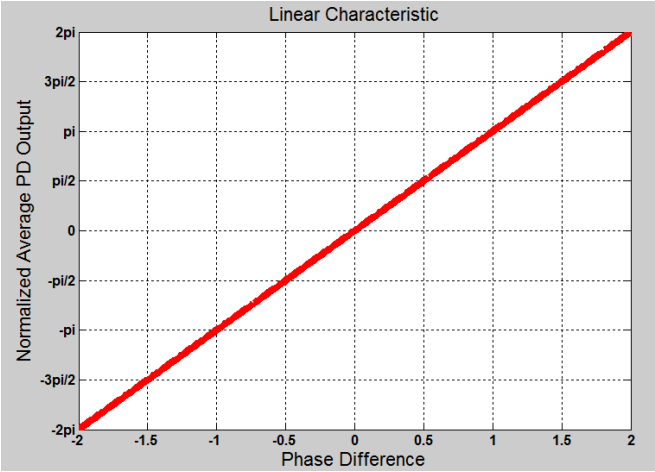


Figure 2.2: Linear PD characteristic

In binary PDs, the output jumps from one extreme to another for almost zero phase difference. Figure 2.3 shows the bang-bang characteristics for ideal and nonideal cases. In the ideal case, the output makes a sharp transition from one extreme to another. However, a finite slope originated from metastability is observed in the realistic characteristic. Metastability occurs whenever clock edge samples data transition points. Note that the input data make a transition from low level to high level and vice versa with a specific rise and fall times respectively. Depending on the application, the type of PD is chosen for clock and data recovery purposes. For low output jitter (i.e., random

variation of the clock edges), linear PDs become attractive. Nevertheless, due to phase offset, the usage of these phase detection devices becomes limited. Binary phase detectors resolve the phase offset problem, but at the cost of relatively high jitter on the recovered clock.

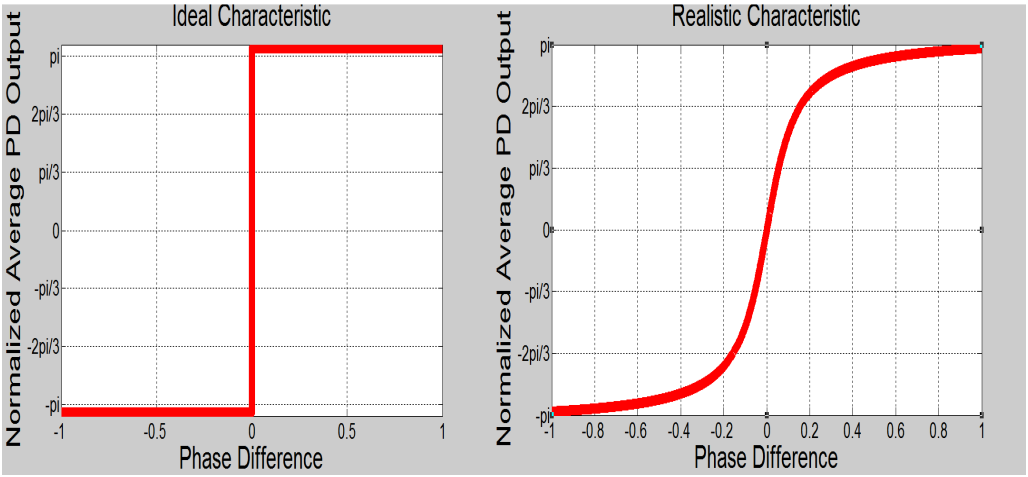


Figure 2.3: Bang-bang PD characteristics

PDs incorporate multiple DFFs in their architecture. Each DFF consists of two back-to-back Latches (Master-Slave). To inspect the effect of metastability on PD outputs, the latch depicted in Figure 2.4 was simulated for three different phase differences between the input data and the clock in 45 nm technology. The latch consists of a differential pair and a cross coupled pair [7]. Clock and $\overline{\text{Clock}}$ signals were applied to NM5 and NM6 transistors. Each half of the circuit was only active for half a clock cycle. The supply voltage was 1 V. An ideal tail current source was employed for transistor biasing. The period of input data and the clock, applied to the circuit, was 2 ns. In the first case, the phase difference between Din and Clk was set to 20 ps. Figure 2.5

demonstrates that the latch outputs (Vx and Vy) made a smooth transition from low level to high level and vice versa. In this case, the differential pair made a complete switching. For the second case shown in Figure 2.6, the phase difference was chosen to be 10 ps. The output difference between high and low levels around the transition intersection was reduced. Eventually by decreasing the phase difference to 7 ps, high and low levels almost touched each other. Figure 2.7 shows this case. Hence, in the metastability region, sometimes outputs do not reach the high (or low) levels, leading to substantial malfunctions in the system performance.

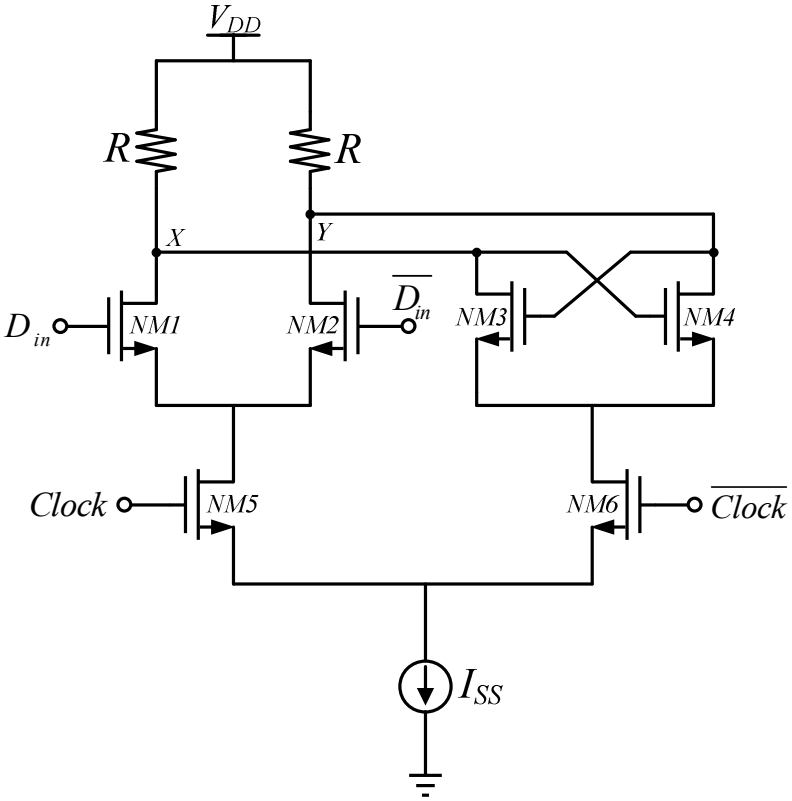


Figure 2.4: Latch utilized in various PD architectures

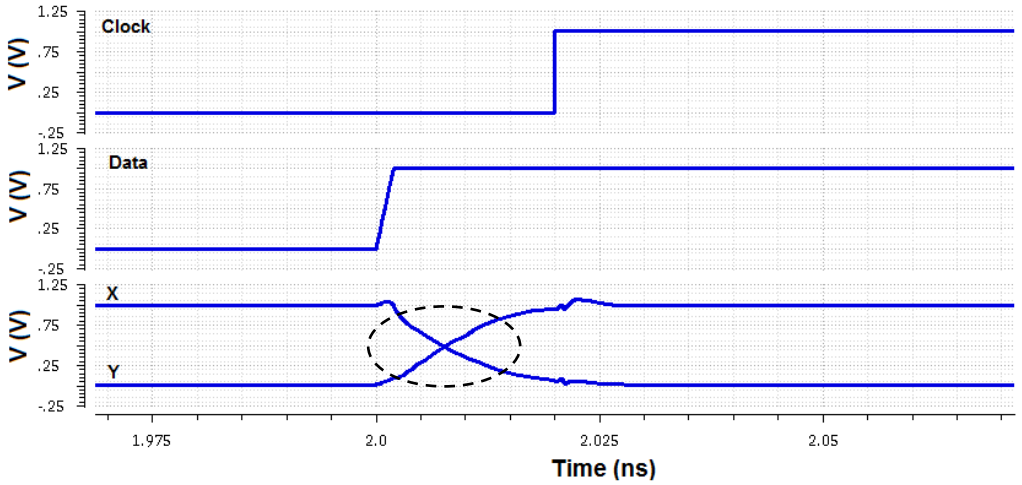


Figure 2.5: PD output for phase difference equal to 20 ps

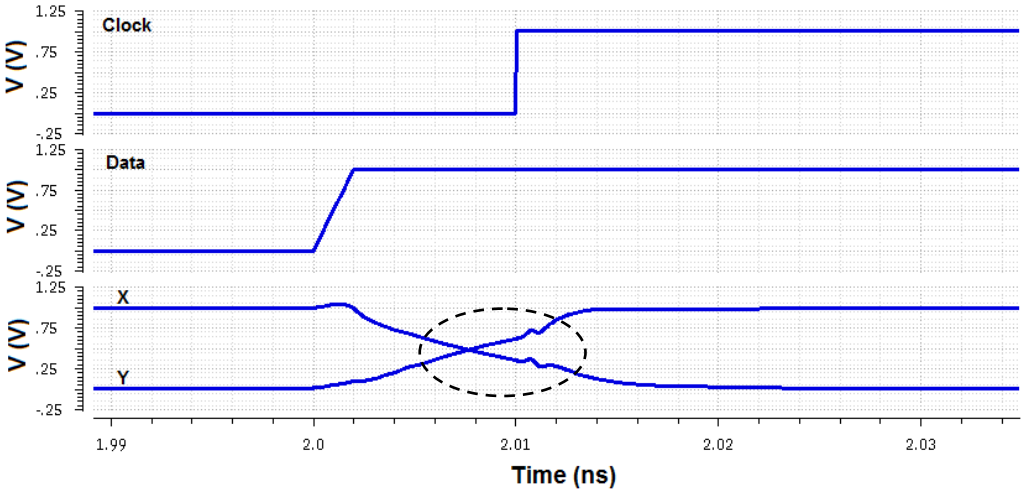


Figure 2.6: PD output for phase difference equal to 10 ps

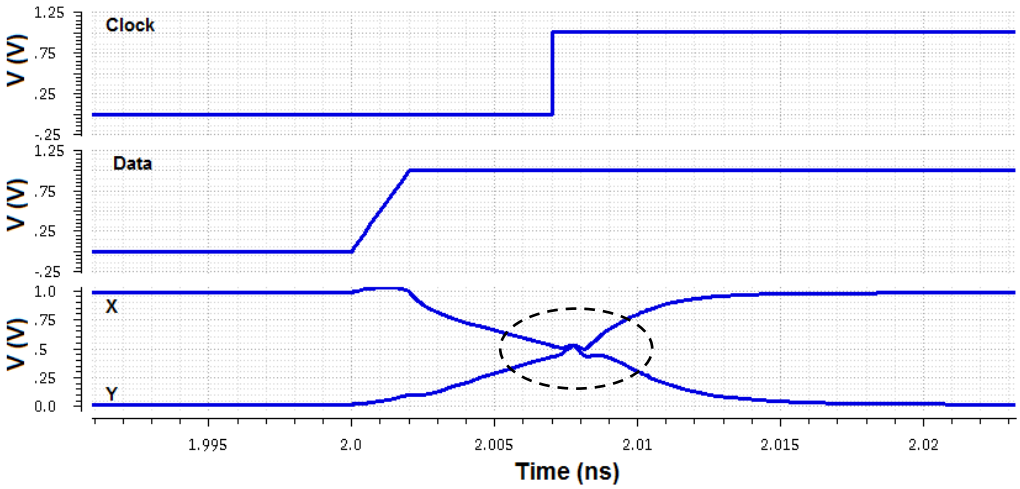


Figure 2.7: PD output for phase difference equal to 7 ps

2.1.2. Charge Pump (CP)

CP is a circuit that charges or discharges the next stage (loop filter) based on the information provided by a phase detector. Basically, a CP consists of two switches and two current sources as shown in Figure 2.8. High data levels at the first output of PDs turn switch S1 on, and as a result, the upper DC current flows through the CP output and charges the output capacitor. Also, high levels at the second output of PDs turn switch S2 on, and eventually the lower DC current flows through the output and discharges the output capacitor. The goal is to provide equal currents flowing into the CP output. However, due to transistor leakage currents, a current mismatch exists between upper and lower parts of CPs. Note that when a transistor turns off, the charge stored in the channel tends to exit through drain and source terminals. Leakage currents arise from these charge injections. To resolve the leakage issue, transistors with high threshold voltages are usually utilized [2].

Another architecture that is currently used in industry for charge pumping purposes is current-steering digital-to-analog converter (DAC). Figure 2.9 demonstrates thermometer and binary portions of an 8-bit current-steering DAC [8]. To reduce nonlinearities and improve the glitch impulse area, eight bits were divided into five bits of segmentation and three bits of binary weight sections. This architecture contains less power dissipation, smaller die size, and more accuracy compared to the other DAC architectures (fully binary weighted and fully segmented). Note that for segmentation part, five bits were converted to thirty-one inputs through a thermometer decoder. A

current mirror system was employed to copy the current provided by the main branch into thirty-one segmented and three binary branches. To minimize the mismatches between consisting devices, thirty-one copy transistors with equal sizes for segmented portion and three copy transistors with doubled sizes for binary portion were utilized. Finally, the outputs of two portions were connected to each other. The charging (or discharging) currents, depending on the values of thermometer and binary inputs, propagate to the output capacitor for the required current generation.

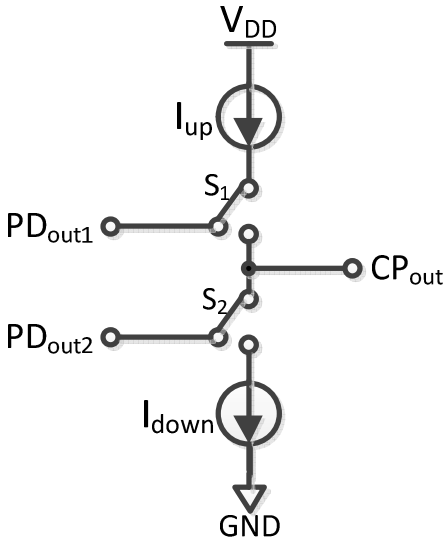
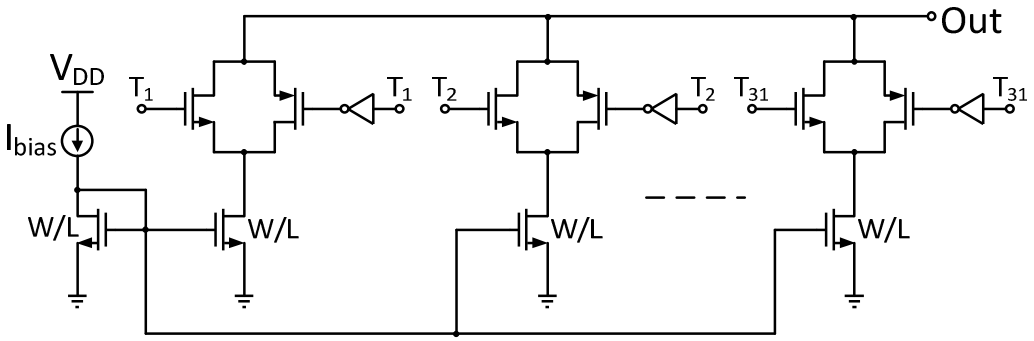


Figure 2.8: Basic charge pump architecture



(a)

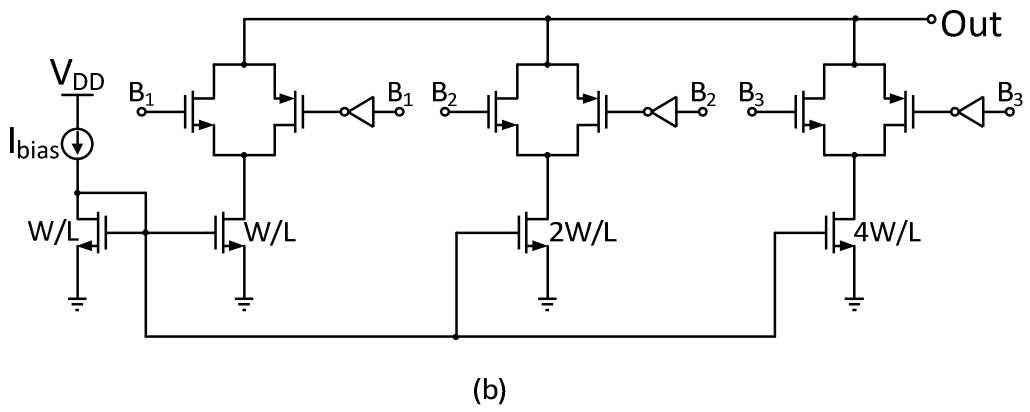


Figure 2.9: Current-steering DAC (a) thermometer portion (b) binary portion

2.1.3. Loop Filter (LF)

Phase detector output contains both low-frequency and high-frequency components. High-frequency components are undesirable and have to be eliminated from the overall information. The best way to perform partial data cancellation is to utilize a filter. A low-pass filter is an ideal choice for high-frequency component elimination. Figure 2.10 depicts the low-pass RC filter schematic. Resistor R contributes to the system stability by adding a zero to the loop transfer function. To suppress the high jumps, due to the charge pump current injection to the filter, capacitor C2 was added to the circuit [2]. Note that the high jumps disturb the system performance severely. The loop filter transfer function is equal to:

$$TF = \frac{V_{out}}{I_{cp}} = \frac{RC_1S+1}{RC_1C_2S^2+(C_1+C_2)S} \tag{2.2}$$

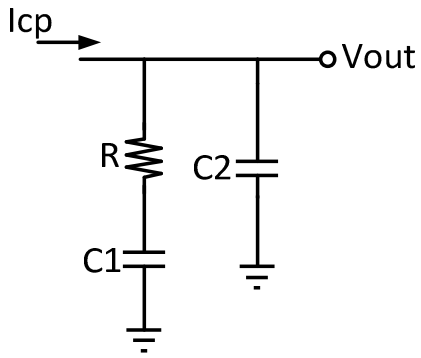


Figure 2.10: Low-pass RC filter

2.1.4. Voltage-Controlled Oscillator (VCO)

VCO is a circuit that generates a clock signal based on the information provided by the low-pass filter. This means that the frequency of the generated clock is varied by the control voltage. Figure 2.11 illustrates the VCO characteristic. Currently, two types of VCOs are being utilized in CDR (or PLL) architectures: Ring and LC oscillators. Ring oscillators possess the capability of generating large tuning ranges ($\omega_1 - \omega_0$). However, the amount of jitter on the recovered clock, compared to LC oscillators, is relatively high [2]. LC oscillators do not produce wide tuning ranges, and consequently designers do not have the luxury of maneuvering in a wide range of frequencies. The relationship between output frequency and the control voltage is described by the following equation:

$$K_{VCO} = \frac{\omega_1 - \omega_0}{V_1} \tag{2.3}$$

Where K_{VCO} is the VCO gain, ω_1 is the maximum oscillation frequency, ω_0 is the free-running frequency (i.e., generated clock frequency with zero control voltage), and V_1 is the VCO control signal. The phase-frequency relationship is resulted by combining

Equations 2.1 and 2.3.

$$\Delta\Phi = \frac{\Delta\omega}{K_{PD}K_{VCO}} = \frac{\omega_1 - \omega_0}{K_{PD}K_{VCO}} \tag{2.4}$$

Figure 2.12 shows examples of ring and LC oscillators. Inverters and LC tanks play a critical role in the generated clock frequency and also CDR (or PLL) performance [4].

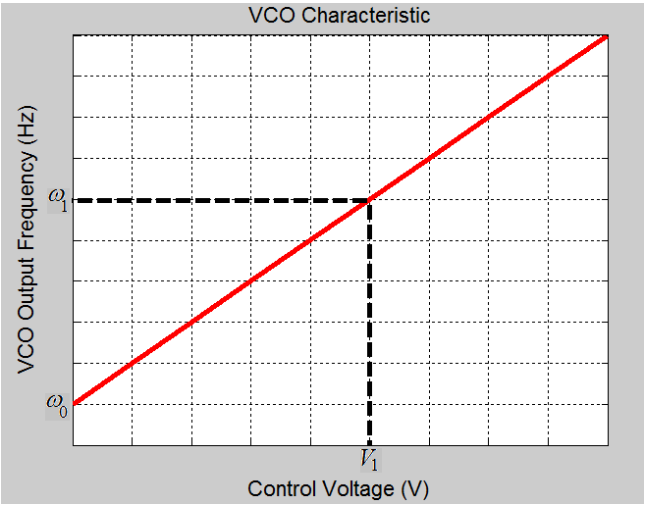


Figure 2.11: VCO characteristic

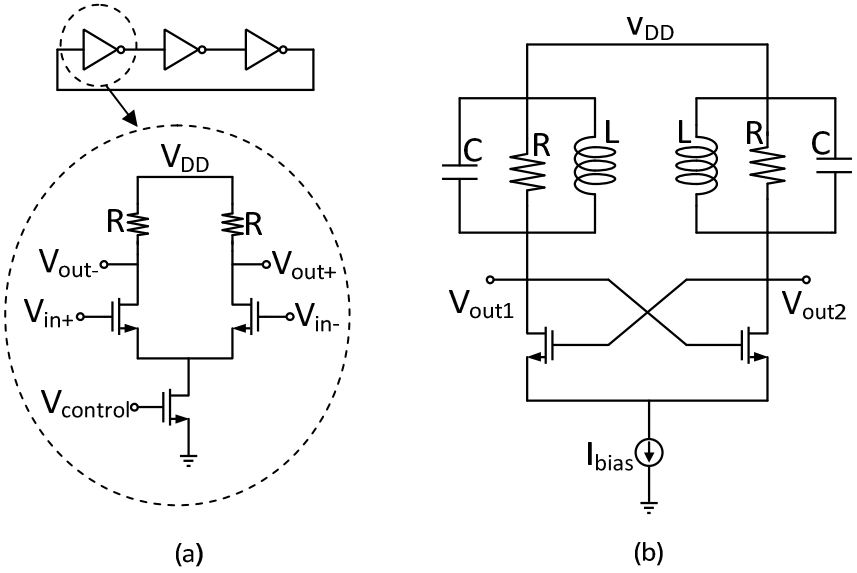


Figure 2.12: (a) Three-stage ring oscillator (b) LC oscillator

2.2. Third-Order PLL

Figure 2.13 depicts a third-order PLL block diagram [2]. The phase detector is presented by a subtractor and a multiplier. The only task of a PD is to convert the phase difference between the input and the output to voltage [2]. The generated voltage is multiplied by the CP coefficient, and the result is filtered out by a low-pass filter. The VCO converts the filtered voltage to the output phase, which is propagated to the PD. Integer-n and fractional-n PLLs incorporate frequency synthesizers to divide the output frequency. The open-loop transfer function of a third-order PLL is:

$$\mathrm{TF}_{open}=\frac{\Phi_{out}}{\Phi_{in}}|_{open} = \frac{K_{PD}I_{CP}K_{VCO}}{2\pi S}\left[\left(R+\frac{1}{C_1 S}\right)||\frac{1}{C_2 S}\right] \tag{2.5}$$

The transfer function contains three poles at the origin. Hence, this architecture is called "third-order PLL." As mentioned earlier, resistor R was added to the circuit to suppress the instability caused by three origin poles. The closed-loop transfer function is shown by Equation 2.6. The frequency synthesizer effect was not considered.

$$TF|_{closed} = \frac{\Phi_{out}}{\Phi_{in}}|_{closed} = \frac{K_{PD}I_{CP}K_{VCO}(1+RC_1S)}{2\pi RC_1C_2S^3+2\pi(C_1+C_2)S^2+K_{PD}I_{CP}K_{VCO}RC_1S+K_{PD}I_{CP}K_{VCO}} \tag{2.6}$$

Note that utilizing two capacitors yields a third-order PLL. This can make the system highly unstable. However, by choosing values from the range of one-fifth to one-tenth of the main capacitor for the second capacitor, PLLs manifest stable behaviors. Moreover,

the value of the loop filter resistor has to be chosen carefully. Large resistors make PLLs unstable [2].

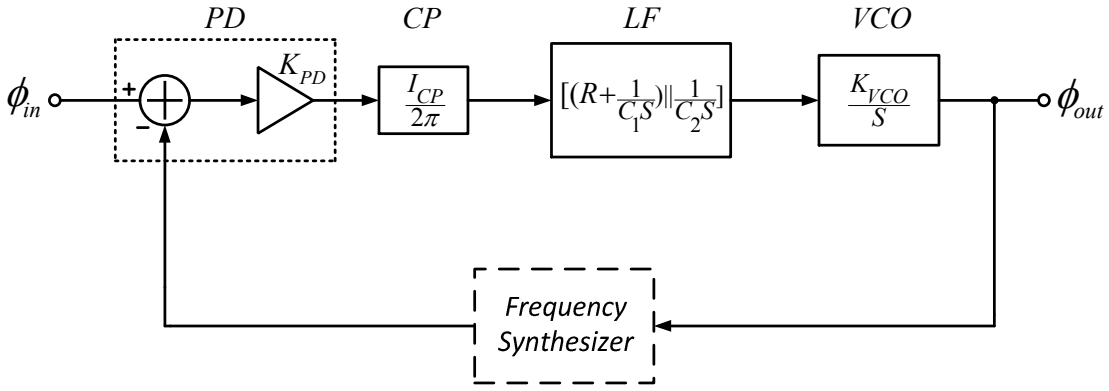


Figure 2.13: Third-order phase-locked loop block diagram

2.3. Jitter in PLL and CDR Circuits

Jitter is random variations of zero crossings of a waveform. Jitter transfer function and jitter tolerance are two important topics in efficient clock and data recovery or phase- locked loop designs. Jitter transfer function is the ratio of the output jitter to the input jitter when the input jitter frequency varies. For low input jitter frequencies, the output jitter is equal to the input jitter. Nevertheless, the input jitter is severely attenuated for high input jitter frequencies. Therefore, PLLs (or CDRs) act like low-pass filters for the input jitter case [4].

VCO also contributes to the jitter on the recovered data and clock. In this case, the low- frequency jitter introduced by a VCO is attenuated, and the high-frequency jitter appears at the output without suppression. Note that the input jitter has to be set to zero.

Hence, PLLs (or CDRs) resemble high-pass filters [2]. Mathematically, the input jitter transfer function for a second-order PLL can be calculated by the following equation:

$$H(s) = \frac{2\xi\omega_n S + {\omega_n}^2}{S^2 + 2\xi\omega_n S + {\omega_n}^2} \tag{2.7}$$

where ξ is the damping ratio which is equal to $\frac{R}{2}\sqrt{\frac{I_{CP}C_1K_{VCO}}{2\pi}}$, and ω_n is the natural frequency which is equal to $\sqrt{\frac{I_{CP}K_{VCO}}{2\pi C_1}}$. For the VCO jitter, we have:

$$H(s) = \frac{S^2}{S^2 + 2\xi\omega_n S + {\omega_n}^2} \tag{2.8}$$

Figure 2.14 depicts the input jitter transfer function ($\frac{\Phi_{out}}{\Phi_{in}}$). The VCO jitter transfer function ($\frac{\Phi_{out}}{\Phi_{VCO}}$) is shown in Figure 2.15.

In highly efficient designs, the amount of jitter peaking observed in jitter transfer graphs should not exceed 0.1 dB. The reason is that when multiple data regenerators are placed in the signal path, the overall jitter peaking would be significant for single jitter peaking of more than 0.1 dB, and consequently the recovered data and clock at the end of chain would not be as clear as expected.

Jitter tolerance is the amount of jitter that can be tolerated by a CDR circuit without increasing BER. Note that BER is the ratio of the number of error bits to the overall number of bits [9]. To find jitter tolerance, a jittered clock with a specific jitter frequency is applied to a CDR circuit. As jitter frequency increases, the clock sampling edges move until they sample different data levels (i.e., if the clock edge was sampling high level, now it starts sampling low level and vice versa). Mathematically, jitter tolerance is calculated by Equation 2.9.

$$G_{JT}(S) = \frac{1}{2}\frac{S^2 + 2\xi\omega_n S + {\omega_n}^2}{S^2} \tag{2.9}$$

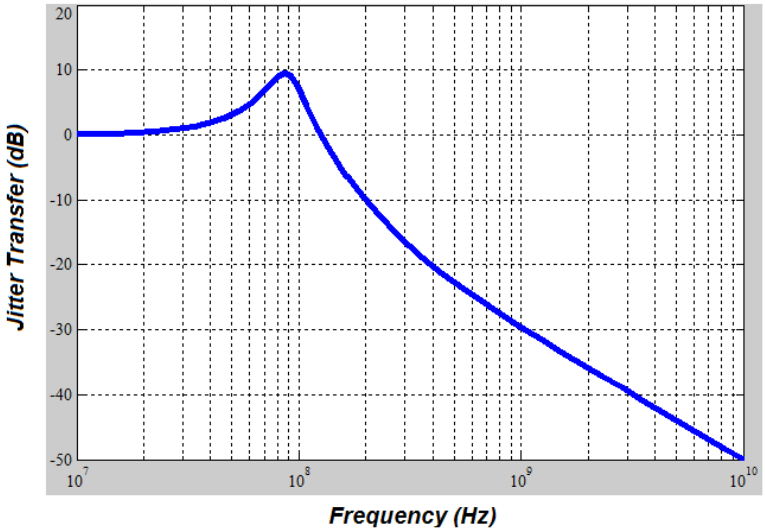


Figure 2.14: Input jitter transfer function

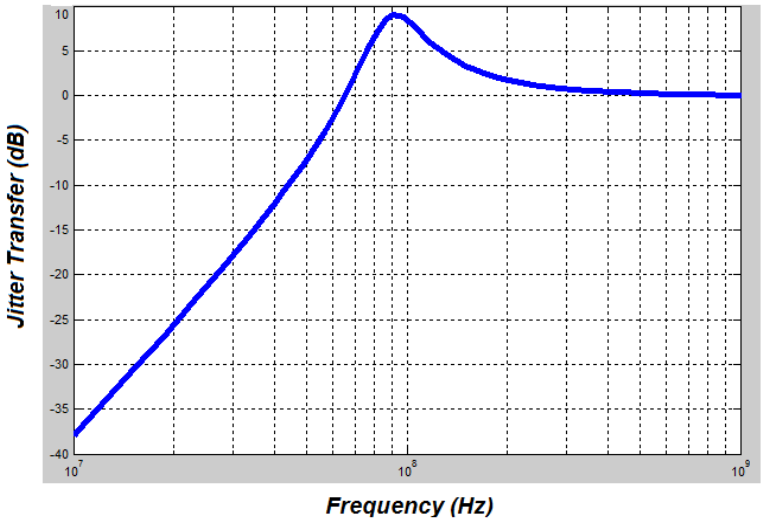


Figure 2.15: VCO jitter transfer function

Jitter tolerance for various ξ values is illustrated in Figure 2.16. The natural frequency was fixed and the damping ratio was increased. Simulations showed that higher ξ led to better results. However, for fixed damping ratio, increasing ω_n resulted in

jitter tolerance improvement. Figure 2.17 depicts jitter tolerance for five different ω_n cases. To conclude, a trade-off exists between jitter transfer bandwidth and jitter tolerance [4].

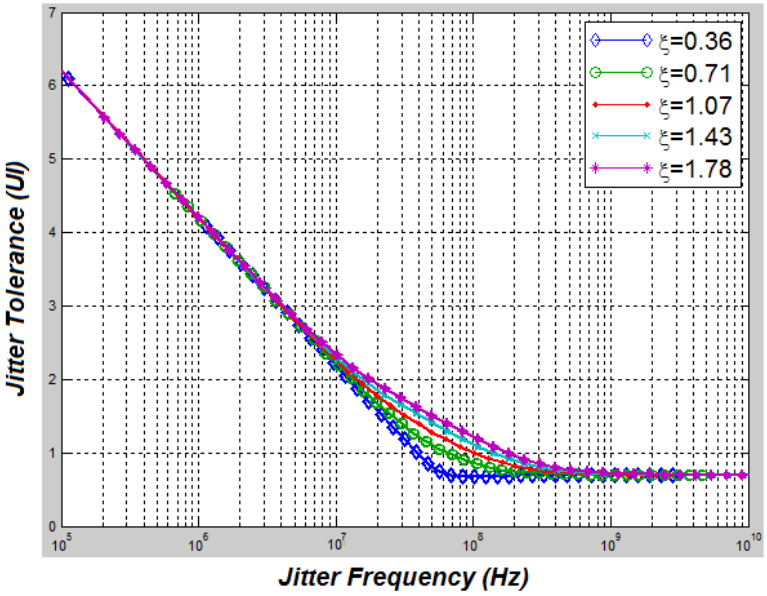


Figure 2.16: Jitter tolerance for different ξ values

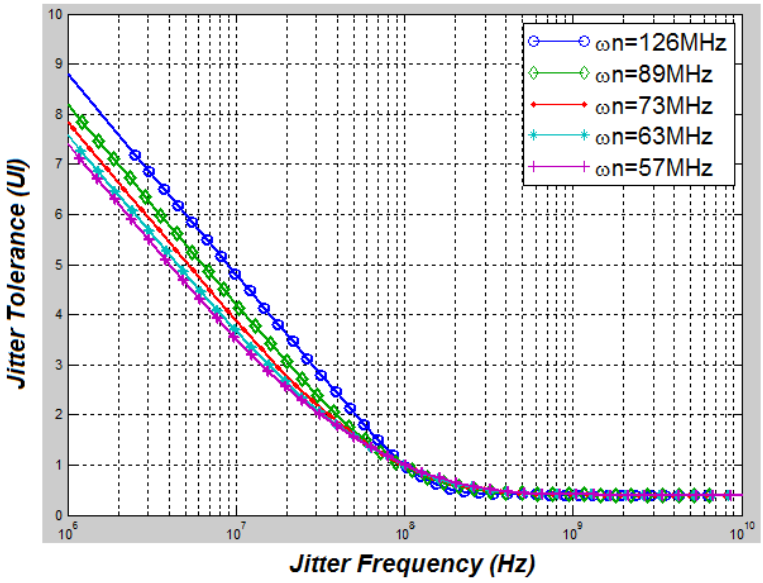


Figure 2.17: Jitter tolerance for different ω_n values

2.4. Delay-Locked Loop (DLL) vs. PLL

DLLs generate various phases of the input signal. Figure 2.18 demonstrates a DLL block diagram [2]. The delay stages (e.g., differential pair) generate various phases by delaying the input signal. The edge spacing between the out_1 and the input signal should be equal to the edge difference between the out_2 and the out_1. The mismatches between delay stages result in different edge spacings. As a result, a combination of PD, CP, and LF is utilized to control the edge difference between generated signals.

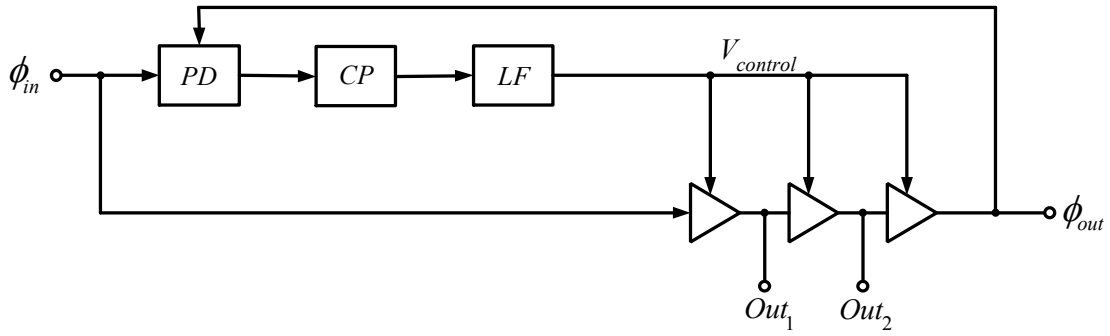


Figure 2.18: DLL block diagram

DLLs suppress noise at the output of delay lines and manifest more stable behaviors. However, this architecture is incapable of producing variable frequencies [2]. Hence, DLLs could not be applied to frequency synthesis applications, which deal with variety of generated clock frequencies. Moreover, to resolve the mismatch issue between delay stages, the devices have to be built larger, raising the area issue. The closed-loop jitter transfer function for a DLL is equal to:

$$TF|_{closed} = \frac{\Phi_{out}}{\Phi_{in}}|_{closed} = \frac{\frac{I_{CP}K_{VCDL}}{2\pi}(RC_1S+1)}{RC_1C_2S^2+\left[C_1+C_2+\frac{I_{CP}K_{VCDL}RC_1}{2\pi}\right]S+\frac{I_{CP}K_{VCDL}}{2\pi}} \qquad (2.10)$$

Where K_{VCDL} is gain of the control line. Figure 2.19 compares the jitter transfer function of PLL and DLL circuits. Note that -3 dB bandwidth of a DLL is much higher than a PLL. DLL jitter tolerance is calculated by the following equation:

$$G_{JT} = \frac{1}{2}\frac{\frac{I_{CP}}{2\pi S C_1}+S}{S(1-e^{-\tau S})} \tag{2.11}$$

Figure 2.20 compares jitter tolerance of PLL and DLL circuits. For the frequencies resided between 4.5 GHz and 22 GHz, the DLL jitter tolerance lies below 0.5 UI.

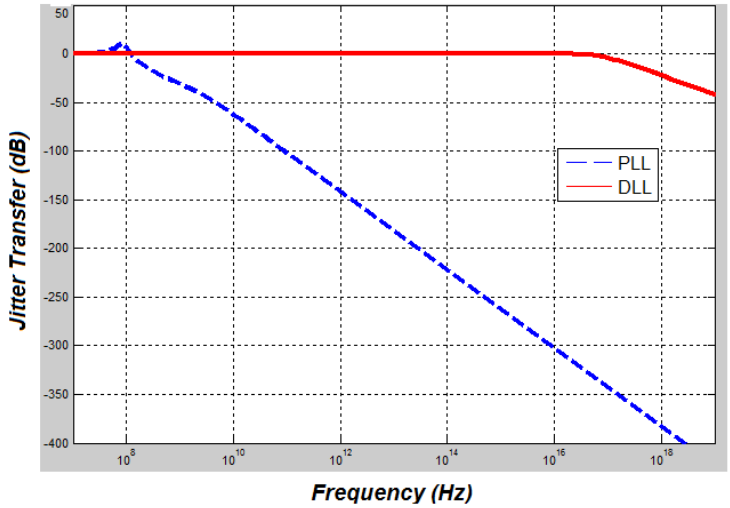


Figure 2.19: PLL transfer function vs. DLL transfer function

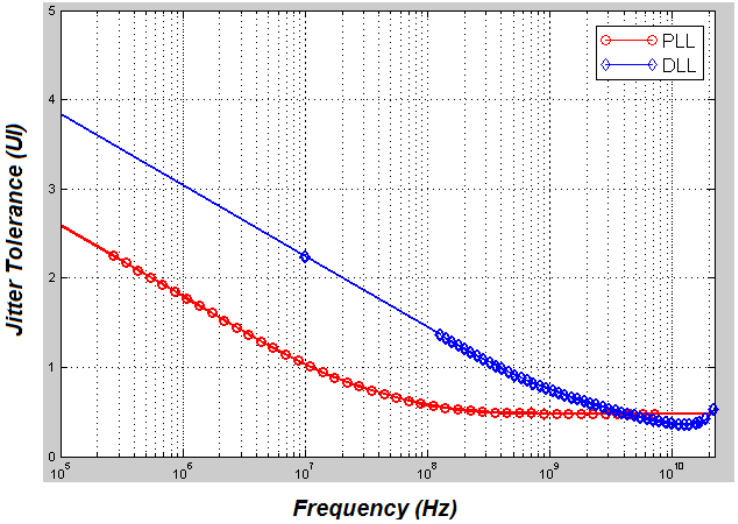


Figure 2.20: PLL jitter tolerance vs. DLL jitter tolerance

2.5. Integer-N and Fractional-N PLLs

Figure 2.21 depicts a typical integer-n PLL. PFD (phase frequency detector), which is a linear PD, is utilized to compare the phase and frequency of the input and output signals. When rising edges at the output of the division block in feedback path occur after reference rising edges, it means that the VCO is running slowly, and we want to force it to work faster. Also, if rising edges of the division output occur before the reference edges, it means the VCO runs fast, and we want to make it perform slower. Note that the feedforward path is continuous time, and the feedback path is discrete time [10]. If we lower the reference frequency by a factor of 10, then M has to be multiplied by 10 to keep the VCO output frequency constant. Decreasing reference frequency provides better tuning resolution, leading to wider range of frequencies. However, reference frequency reduction decreases the PLL bandwidth, which should be smaller than $fref/10$, introducing larger VCO noise (i.e., less VCO noise will be filtered out). Furthermore because of multiplication at the feedback path, larger in-band divider, reference, and charge pump noise will be obtained. Therefore, due to these limitations, integer-n PLLs are used only at specific situations [10].

The feedback part of the integer-n PLL divides the output frequency and compares the resulted frequency with the reference. Figure 2.22 shows the modeled frequency synthesizer for the integer-n PLL. DFFs are being utilized in contemporary frequency divider models. The proposed model was built by using switch, memory, sum, hit crossing, constant, and XOR blocks. The division switch threshold is calculated by

subtracting the division number by 0.5. The sampling time of two back-to-back delays is different. Figure 2.23 shows examples of frequency division.

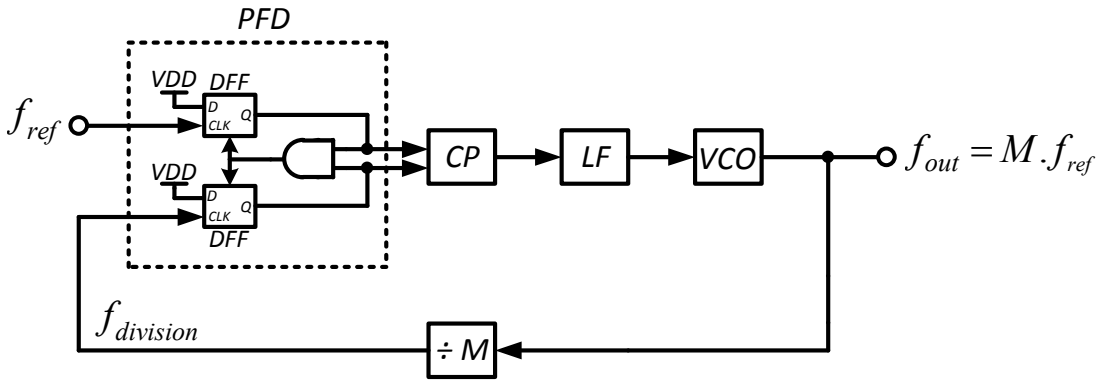


Figure 2.21: Integer-n PLL block diagram

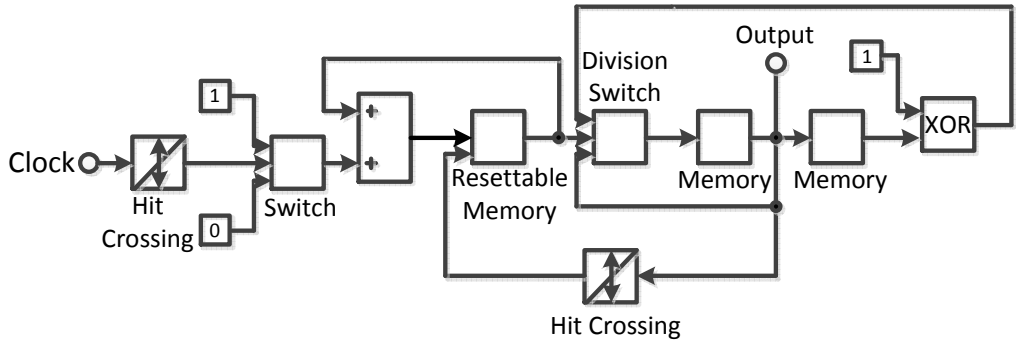


Figure 2.22: Integer-n PLL frequency synthesizer model

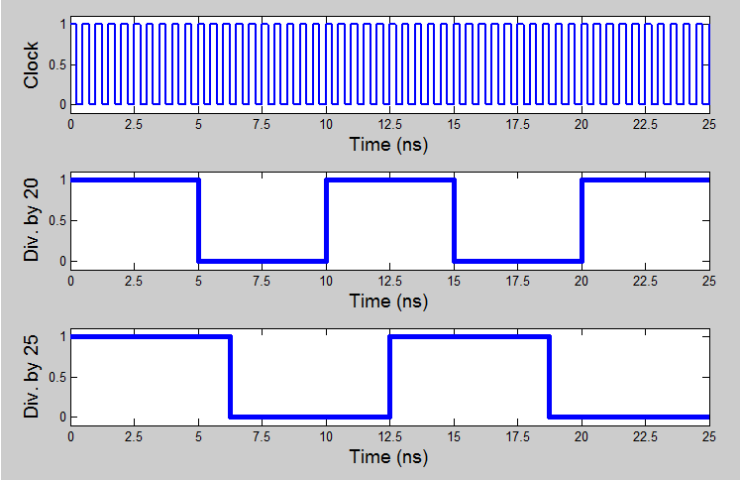


Figure 2.23: Frequency division examples

A fractional-n PLL block diagram is shown in Figure 2.24. The reference frequency and the VCO output frequency were set to 19.68 MHz and 2.403 GHz respectively. The division number was equal to 122.1036585. The division block can only evaluate integer values, and there is no possible way to put the calculated value in the division block. Therefore, the idea would be to switch periodically between 122 and 123 so that the average modulus is equal to 122+51/492. To generate 51/492, we could use shift registers. However, the best method for such purposes is to utilize digital ΔΣ modulator. The advantages of this method are: First, we get the desired VCO frequency, and second, the PLL bandwidth is up to 2 MHz. Yet, this architecture experiences spurious tones due to periodic switching. These tones are not suppressed at low frequencies. Even decreasing bandwidth will not be practical because the goal of this architecture is to increase the loop bandwidth. Note that since every delta-sigma modulator uses a quantizer in its structure, the modulus will contain quantization noise. In well-designed delta-sigma modulators, quantization noise is suppressed, so that it will have zero-mean, and most of its power will be placed outside the PLL bandwidth. Higher-order DSMs diminish the quantization noise, but are rarely used due to PLL filtering limitations [10].

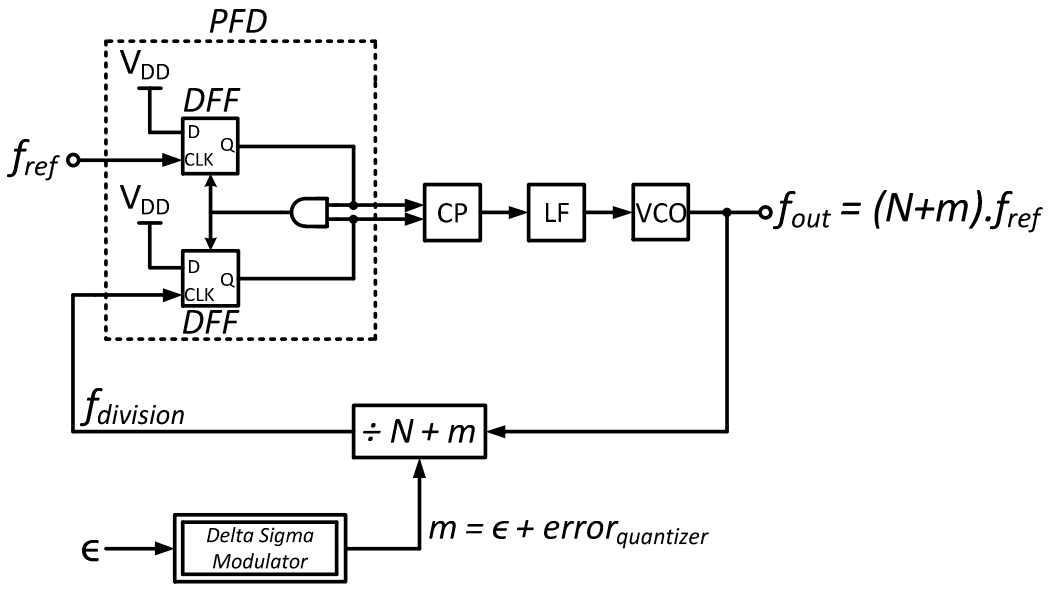


Figure 2.24: Fractional-n PLL block diagram

2.5.1. Second-Order Delta-Sigma (ΔΣ) Modulation

Delta-sigma modulation plays a critical role in fractional-n PLL circuits. Hence, understanding the impact of DSM on the overall system and also getting familiar with advantages and disadvantages of this method assist us in efficient PLL designs. Figure 2.25 demonstrates a second-order DSM test-bench. In the first case, a sine wave with a frequency of 45 KHz and a noise signal with the upper and lower noise bounds of 0.1 and 0 respectively were summed and applied to the system in the absence of DSM [10]. The sampling frequency was 45 MHz. A low-pass filter with the cut-off frequency of 450 KHz was provided to filter out unwanted harmonics. Figure 2.26 depicts the output of the quantizer and the low-pass filter. The quantizer received a sine wave in continuous mode and converted it to a sine wave in discrete mode. I expected to have a clear sine

wave at the output of the filter. However, the result did not meet the expectation. To further inspect the system performance, the power spectral density of the quantizer and the filter outputs were plotted. Figure 2.27 shows the results. Simulations showed that harmonics located at the right-hand side of 450 KHz were filtered out. Yet, unwanted harmonics still existed in the left-hand side, causing the output waveform not to be sinusoidal.

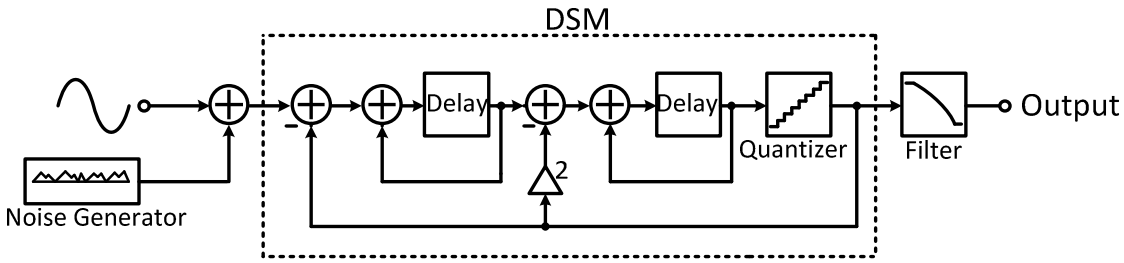


Figure 2.25: Second-order DSM test-bench

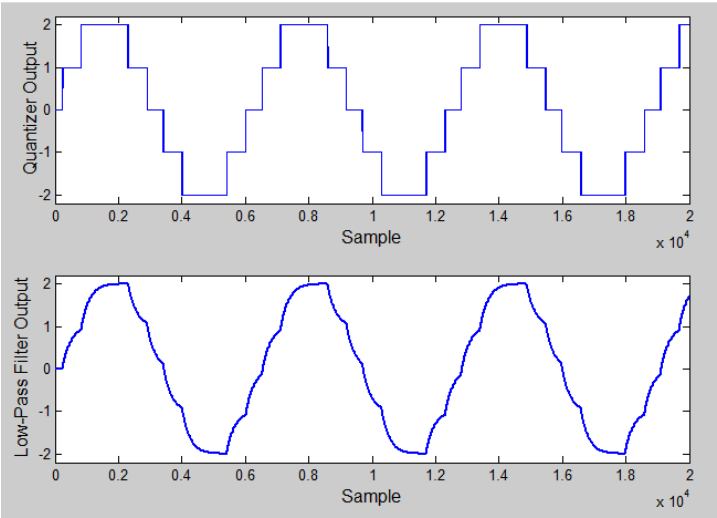


Figure 2.26: Quantizer and filter outputs in the absence of DSM

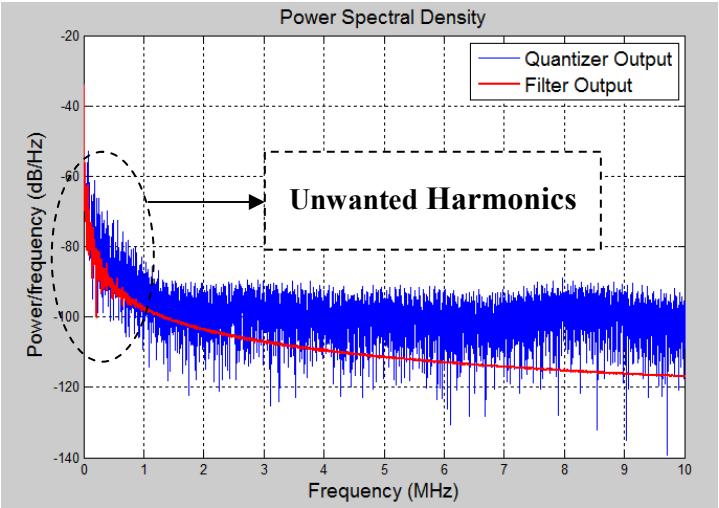


Figure 2.27: Quantizer and filter PSD in the absence of DSM

In the second case, a second-order DSM was interposed between the noisy input signal and the loop filter [10]. The DSM consists of delay, adder, subtractor, multiplier, and quantizer blocks. Note that we are dealing with a digital delta-sigma modulator, which generates integer values such that the average of 122 plus the values produces the feedback division number. Figure 2.28 shows the outputs of the DSM and the low-pass filter. The noisy waveform at the output of the DSM was considerably cleaned by the filter, and the result was a very clear sinusoidal signal. From frequency aspect, as shown in Figure 2.29, the unwanted high-frequency components were eliminated, and only fundamental harmonic was present. Therefore, by utilizing the DSM methodology, the unwanted harmonics resided in the pass-band of the filter were pushed into the stop-band of the filter. The DSM technique is extensively utilized in high-quality data conversion (ADC or DAC) projects, which require high linearity and accuracy.

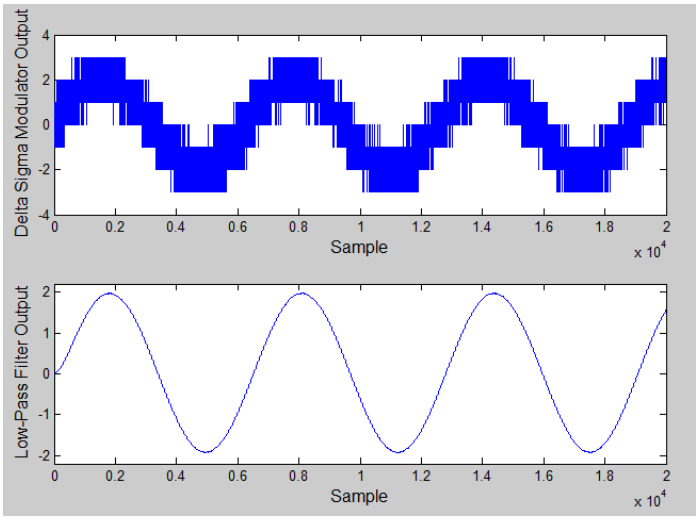


Figure 2.28: DSM and filter outputs

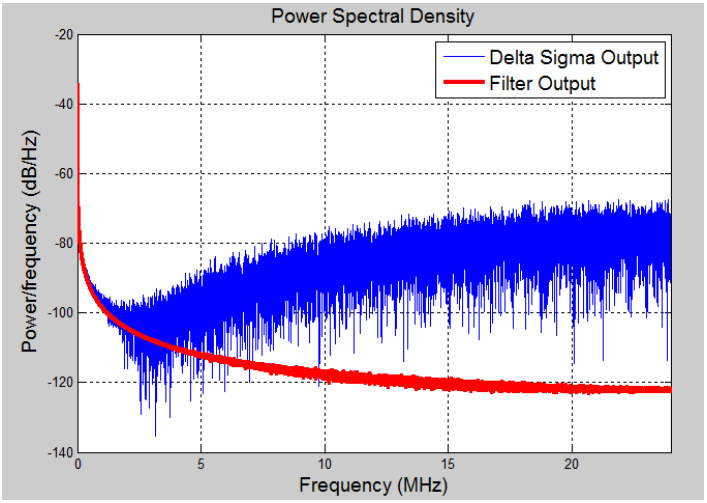


Figure 2.29: DSM and filter power spectral densities

2.5.2. Fractional-N PLL Design

In this section, a fractional-n PLL (shown in Figure 2.25) design methodology is explained. The reference frequency provided by a low-noise crystal oscillator and the output frequency generated by the VCO were equal to 19.68 MHz and 2.403 GHz

respectively. Loop filter components (i.e., R, C1 and C2) were equal to 1 KΩ, 20 nF, and 1 nF respectively. Figure 2.30 illustrates the feedback path of the designed fractional-n PLL. The frequency synthesizer block provides the required division number (integer + fraction) and generates a clock signal based on the obtained frequency. Each rising edge of the generated division clock triggers the DSM to produce an integer value for the frequency synthesizer. Note that in this particular case, the division number was equal to 122.1036585. The clock frequency at the output of the VCO is monitored, and the produced fractional value is passed through the DSM to generate required integer values. The fractional value has to be added to a pseudo-random sequence, which is called "Dither," to cancel spurious tones [10].

Phase margin and bandwidth of the designed PLL can be mathematically calculated by the following equations:

$$\text{PM} = \frac{180 \times acrtan(\frac{b-1}{2\sqrt{b}})}{\pi} \tag{2.12}$$

$$f_{BW} = \frac{I_{CP} K_{VCO} R}{2\pi M} \times \frac{b-1}{b} \tag{2.13}$$

Where b is the ratio of C1/C2 plus one, and M is the division number, which in this case was 122.1 [10]. The charge pump current and the VCO gain were 180 μA and 250 MHz/V respectively. Hence, the designed system contained a phase margin of 65.9 deg and bandwidth frequency of 55.7 KHz.

PLL phase noise is the summation of reference crystal oscillator, PFD, CP, LF, VCO, and divider phase noises. The result is summed with delta-sigma modulator

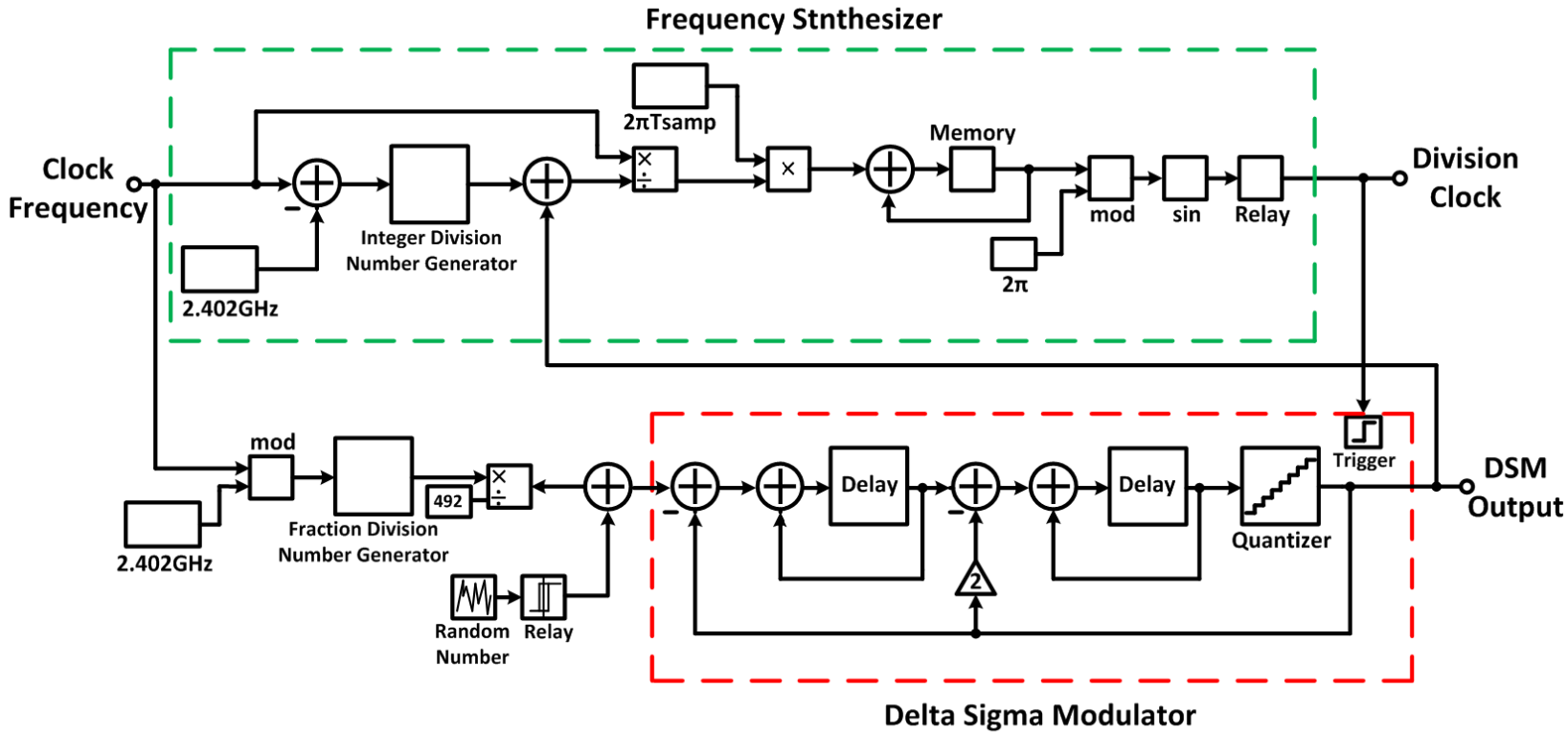


Figure 2.30: Feedback path of a fractional-n PLL

quantization noise. PLL phase noise due to only ΔΣ modulation is calculated by the following equation:

$$Phase\ Noise|_{\Delta\Sigma} \approx 10\ \times \log\left[\frac{4\pi^2 b}{3f_{ref}} \times sin^2\left(\frac{\pi f}{f_{ref}}\right) \times \left(\frac{f_{BW}}{f}\right)^2\right] \tag{2.14}$$

As a result, the PLL phase noise due to the modeled delta-sigma modulator was approximately equal to -131 dBc/Hz. Figure 2.31 depicts the linearized and the actual simulation results for the designed PLL [10].

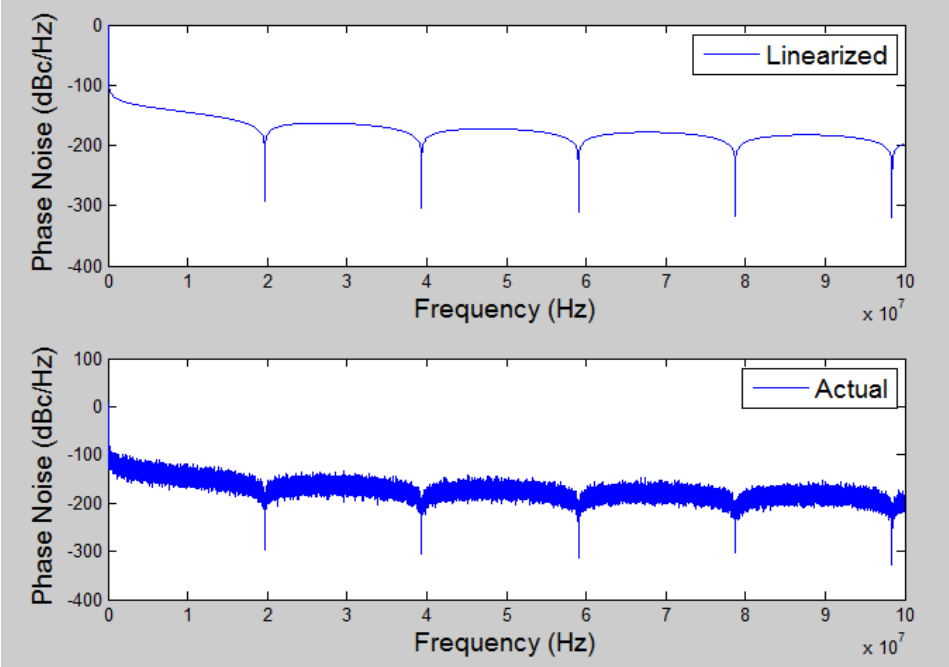


Figure 2.31: PLL results

Chapter 3. CDR Matlab and Simulink Models

3.1. DFF Metastability Model

Delay flip-flop modeling was performed in Simulink by using standard blocks such as switch and unit delay. The range of setup, hold, and C2Q times is determined by the sampling time [1]. Table 2 provides information for DFF timing metrics. As mentioned before, C2Q time must be greater than zero. Otherwise, delay flip-flop will act like a wire, and there would be no need to place a DFF in the model [2]. The minimum value for propagation delay from the clock signal to the output is equal or greater than T_{samp}.

Table 2. Minimum value and range of timing parameters

	Minimum Value	**Range**
$\mathbf{T_{su}}$	0	Greater than T_{samp}
$\mathbf{T_{ho}}$	0	Greater than T_{samp}
$\mathbf{T_{cq}}$	T_{ho} and greater than 0	Greater than or equal toT_{samp}

The main blocks in the modeled DFF are:

- Positive trigger
- C2Q time counter
- Setup time counter
- Hold time counter
- Violation and data sampling
- Metastability path
- Lock subsystem

Figure 3.1 depicts the implemented DFF in Simulink. Each block will be explained in the following subsections in details.

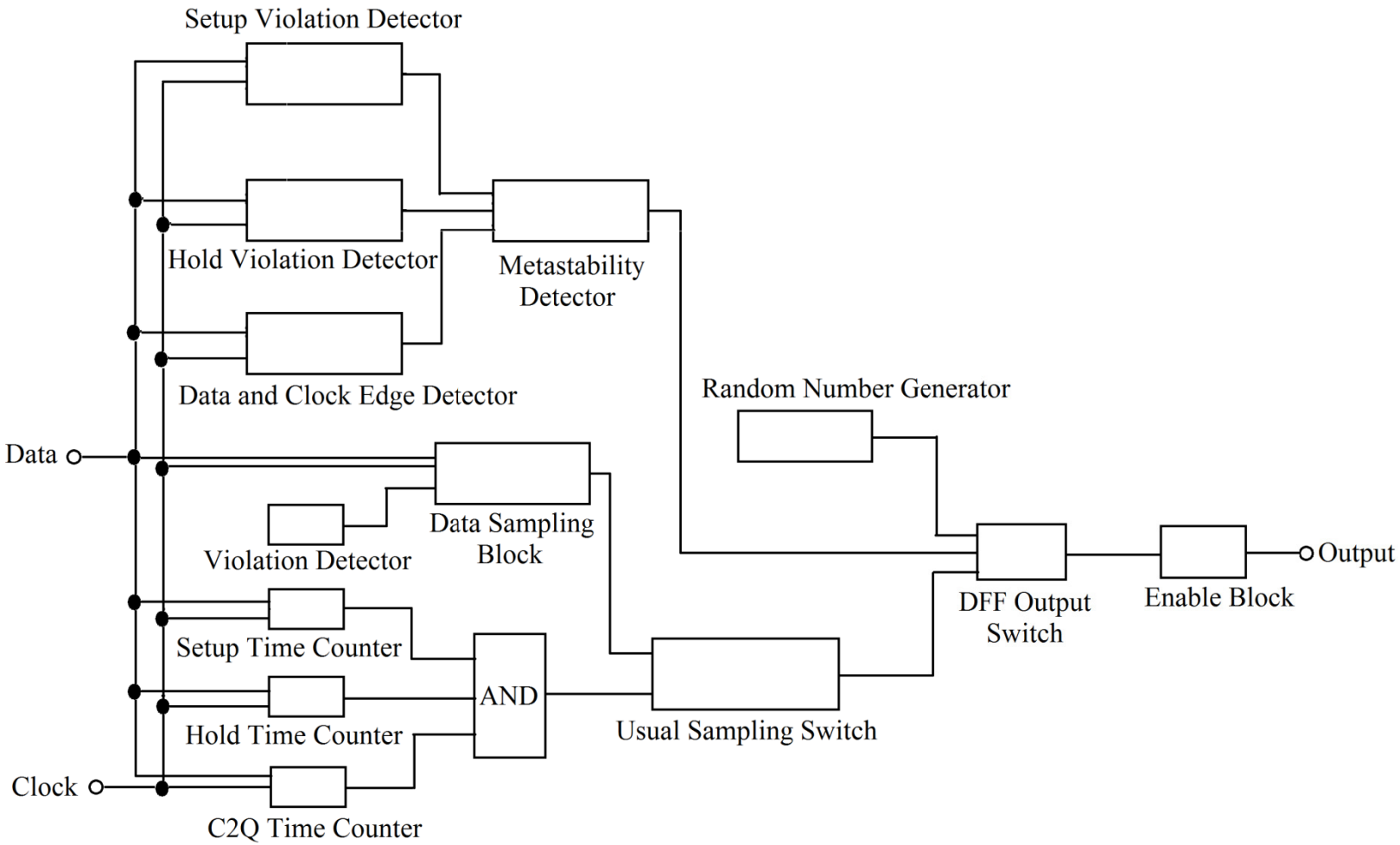


Figure 3.1: DFF metastability model in Simulink

3.1.1. Positive Trigger

When positive clock edges occur at the input, the output of this block is set to high. The width of the trigger pulse is equal to the sampling time. The purpose of generating such narrow pulses is to inform the circuit that positive clock edges have been observed, and we want to prepare the circuit to calculate the timing parameters. The narrow pulses will be generated by using an XOR with the clock signal and the delayed replica of the clock signal at the input and a switch with an appropriate threshold at the output to filter out only rising narrow pulses.

3.1.2. C2Q Time Counter

The output of the previous stage was applied to this block. First, the narrow pulses reset the block. This means with rising edges of the clock, we start counting C2Q time. Then, T_{C2Q} is compared with T_{samp}. If T_{samp} becomes equal to T_{C2Q}, then the output of the counter will be set to high, meaning C2Q time requirement has been fulfilled. Otherwise, the comparison will be resumed by using a loop until the two values become equal.

3.1.3. Setup Time Counter

Due to input data levels (high and low), two different paths were provided in this subsystem. One path was allocated to high data levels, and the other one to low data levels. When a high data level is applied to this block, the upper path will be activated, and T_{samp} will be compared with T_{su}. If T_{samp} is equal to T_{su}, then the output of the switch

will be high. Otherwise, we would expect to have low level at the output, meaning the counting function is still carrying on until it reaches the target (T_{su}). Besides, when a rising edge of the clock is present, setup counting function has to be stopped, and the result, which determines the timing requirement has been met or not, should be revealed. Therefore, a lock subsystem was provided to fulfill this requirement. The lock subsystem stops counting and propagates the counting result to the output of the setup time counter.

3.1.4. Hold Time Counter

Whenever a rising edge of the clock occurs at the input, the hold time counting should start. To implement, two different subsystems were provided. One of them resets the counting on both data transitions and positive clock edges, and the other one only resets the counting on the rising edges of the clock. This means the output of the second block is set to high either before the first one or at the same time. As mentioned before, the maximum allowable value for hold time is the time interval between the sampling edge of the clock and data transition points. So, the goal is to count the hold time during the mentioned time interval. The outputs of the two subsystems were applied to an AND gate and finally propagated to the output. In this case, we used the lock subsystem because counting has to be stopped at data transitions, and the result must be revealed.

3.1.5. Violation and Data Sampling

The violation block was modeled according to Table 1. If one of these conditions had not been fulfilled, DFF output would have been zero. In other words, this block acts

like a switch. When all timing metric requirements are met, the switch will be closed, and the data will proceed to the subsequent stages. The violation block was connected to the data sampling block in which the incoming data were sampled by the clock signal. In fact, the violation block plays the enable port role in the data sampling block.

3.1.6. Metastability Path

When the clock rising (or falling) edges sample data transition points, DFFs generate random outputs (i.e., sometimes zero and sometimes one). This behavior is caused by the metastability and has a great impact on the system performance especially in transistor-level designs. To model this behavior, data and clock transition points were modeled to be monitored by edge detectors. If data and clock transition points happen at the same time or timing violation occurs, a random number propagates to the DFF output. This is exactly the behavior observed in transistors.

3.1.7. Lock Subsystem

The lock subsystem was utilized in two blocks: setup time counter and hold time counter. The reason we use lock subsystem is to stop the counting action and observe the result. For example, the lock subsystem used in the setup time counter has two inputs: positive trigger clock and C2Q enable. Whenever positive trigger clock is set to high, the lock enable becomes activated. When positive trigger clock becomes low, and C2Q enable is set to high, the output will be disabled. The enable block was also provided for linear phase detection purposes (e.g., fractional-n PLLs). The block was interposed

between the DFF output switch and the output port. The signal at the switch output was ANDed with the input enable signal. If the DFF becomes activated, the sampled data will propagate to the output port. Each rising edge of the clock resets the enable block.

3.1.8. Enable Block

For integer and fractional-n PLLs, phase-frequency detectors are utilized. In such applications, beside data and clock ports of DFFs, the reset port is actively involved in phase and frequency detections. Hence, an enable block was provided prior to the DFF output. When the enable block is disabled, the output is set to zero. Otherwise, the processed data are propagated to the output.

3.2. DFF Metastability Testing

To test the modeled DFF, a simple test-bench was prepared. A high-level data signal with the width of $T_{su} + T_{ho}$ was generated. The goal is to show how flexible the DFF acts when a timing metric violation occurs. The DFF output will only have high level if the clock rising edges sample the midpoints of high-level data. Any other sampled point will not fulfill the requirements.

Figures 3.2, 3.3, and 3.4 depict the simulations for three different clock sampling points. Setup and hold times were equal to 0.1 s, and C2Q time was equal to 0.2 s. The sampling time was set to 0.01 s. Therefore, the input data were high for 0.2 s, which started at 10 s and ended at 10.2 s. The second graph in Figure 3.2 shows the rising edge of the clock occurred at the midpoint of the data eye (i.e., 10.1 s). Eventually the DFF

output became high after 0.2 s due to C2Q time. In Figure 3.3, the clock rising edge was pushed forward to sample the input data at 10.101 s. It means the clock edge was moved by + δ, which is equal to the sampling time. The third graph in Figure 3.3 shows the result. Also, if the clock sampling edge moves backward by – δ, the result will look like the previous case. Figure 3.4 illustrates the clock sampling edge occurred at 10.099 s and the DFF output. Overall, the results indicated that the DFF metastability manifests proper results when it is encountered by any timing metric violation. Various setup, hold, and C2Q times were examined, and the results were all convincing.

To ensure that the DFF functions properly, another test-bench was provided by utilizing a pulse generator with an amplitude of 1 V and frequency of 2 Hz as a data stream and a manual clock generator block as the clock source. The clock generator block incorporates ten step blocks, five XOR blocks, and a sum block. Every two consecutive step blocks were XORed to generate a narrow pulse, which plays the role of sampling edge of the clock. Eventually five narrow pulses were generated at 4.2 s, 24.7 s, 37.7 s, 68 s, and 80 s. These times were randomly selected to observe the functionality of the DFF at both counting and metastability paths. Setup, hold, C2Q, and sampling times of the DFF were equal to 0.1 s, 0.1 s, 0.2 s, and 0.025 s respectively. At 4.2 s, the clock sampled a high level of the data, and the DFF output became high. At 24.7 s, the clock sampled a high level again, and the DFF output sustained the previous state.

At 37.7 s, the clock sampled a low level of the data, and the DFF output fell to zero. Figure 3.5 depicts the input data, the clock signal, and the DFF output. To inspect the performance of the DFF at metastability points, two data samplings occurred at 68 s

and 80 s. At these points, the clock signal sampled the data transition points, and the DFF entered metastability. It means the DFF output could not be predicted, and as a result, a random value would be propagated to the output. To model this behavior, a random value generator was placed in the metastability path. Figure 3.6 demonstrates the random data generator output. At 68 s, a low-level data appeared at the DFF ouput. However, at 80 s, the output was switched to high level because of the generated random value.

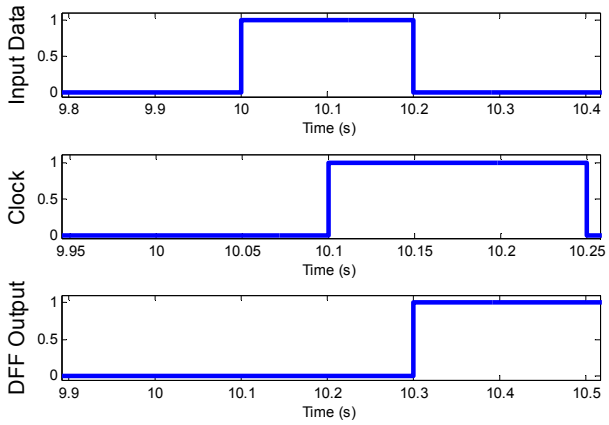


Figure 3.2: DFF simulation result when clock samples the middle of data eye

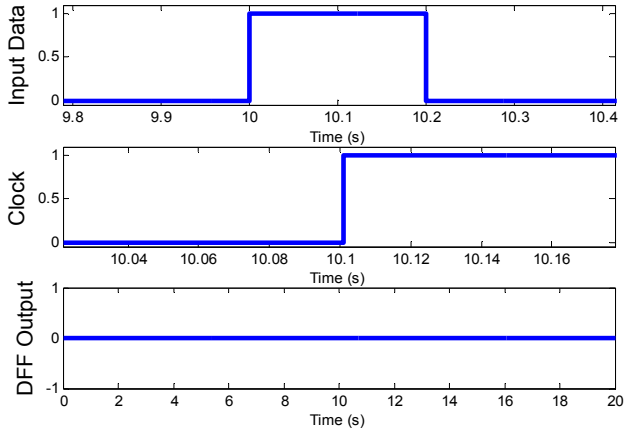


Figure 3.3: DFF simulation result when clock moves forward by + δ

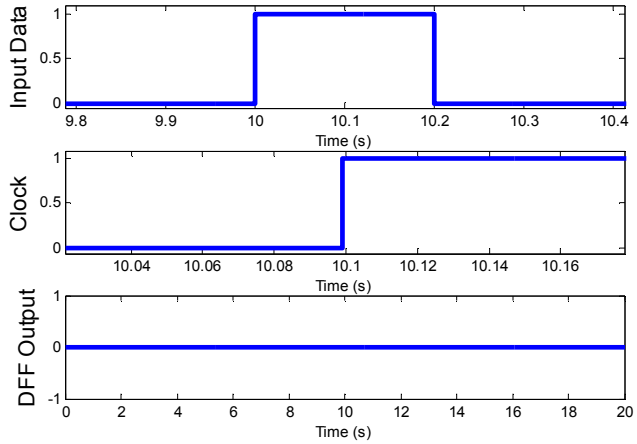


Figure 3.4: DFF simulation result when clock moves backward by – δ

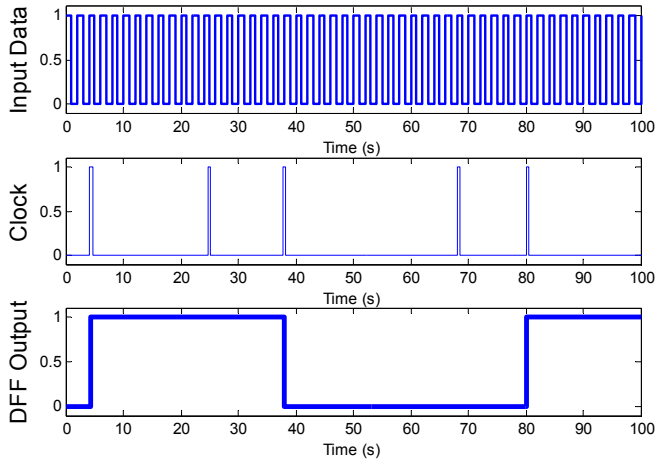


Figure 3.5: DFF metastability simulation result when clock samples data edge

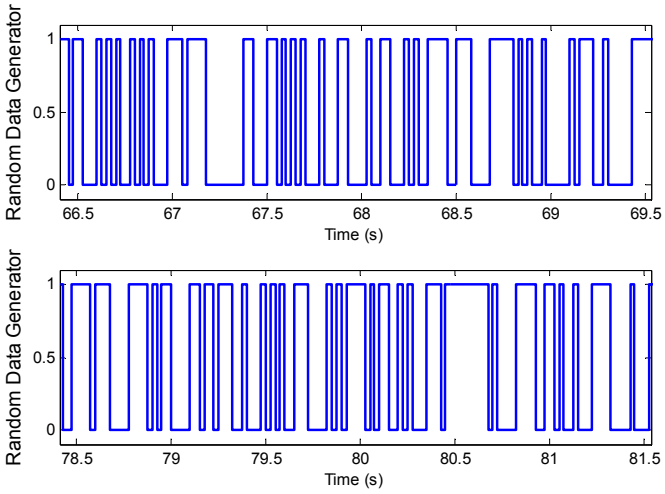


Figure 3.6: Random data generator output when clock samples data edge

3.3. CDR Modeling Using DFF Metastability

Other sets of simulations could be performed by utilizing the implemented DFF in an Alexander phase detector structure. The phase detector was used in the CDR model of Figure 3.7 [11]. An Alexander phase detector is presented by a subsystem. The subsystem incorporates four modeled DFFs and two XOR gates. The early and late outputs of the phase detector drive a voltage-to-current converter. The V/I would charge or discharge the capacitors of the next stage (loop filter). When early output is high, the up switch is on, and current is injected to the loop filter, charging the capacitors. When late output is high, the down switch is on, and the current source will discharge the loop filter capacitors. Ideally, up and down current sources are equal. The amount of current injection by the V/I was 200 μA / 2π, which was approximately equal to 32 μA. Figure 3.8 (a) shows the V/I model. Moreover, as illustrated in Figure 3.7, the loop filter consists of a resistor and a capacitor in series and a second capacitor in parallel with them. The CDR filter was modeled based on the loop transfer function (Equation 2.2). This configuration was implemented in Simulink by using sum and gain blocks. The values of R, C1, and C2 were equal to 1 KΩ, 4 pF, and 400 nF, respectively. Figure 3.8 (b) shows the loop filter model. The VCO was modeled according to the output frequency equation shown by Equation 2.3 [12]. The output of VCO was fed to the clock port of Alexander phase detector. The values of free running frequency and VCO gain were equal to 10 GHz and 1 GHz/V, respectively. Initially, PRBS7 (i.e., $2^7 - 1$) 10 Gbps generated a data stream, and then at 0.2 μs, PRBS7 9.5 Gbps took over by using a switch.

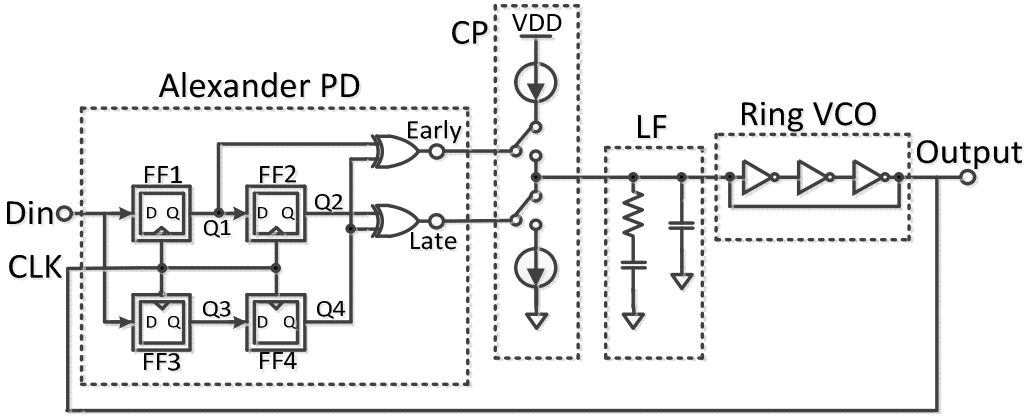


Figure 3.7: CDR model in Simulink and Cadence Virtuoso

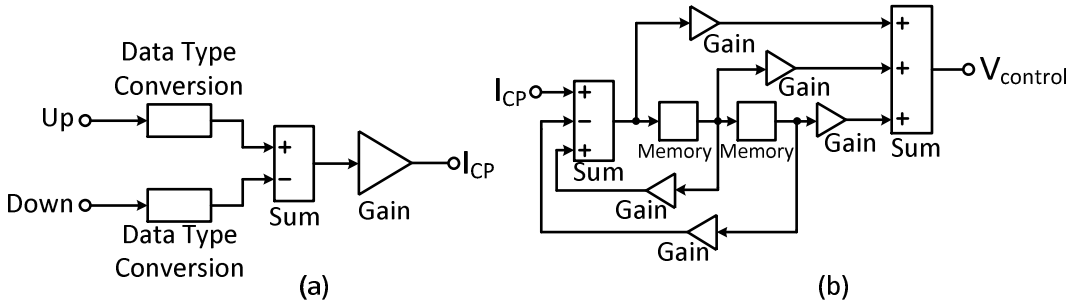


Figure 3.8: (a) CP (b) LF models

For CDR simulation, the data and clock frequencies were equal to 5 GHz and 10 GHz respectively. The sampling time was equal to $T_{clk}/100$. Figures 3.9, 3.10, 3.11, and 3.12 demonstrate the VCO control voltage for various values of setup, hold, and C2Q times. The sampling time was set to 1 ps for all cases. The setup and hold summation was constant for all cases. By adding buffers in the path of data and clock signals, setup and hold times could be regulated.

Table 3 illustrates the results for four different cases. The summation of setup and hold times for three different cases was equal to 20 ps, and C2Q time was the same for three cases and equal to 20 ps. The fourth case had the summation of 2 ps and C2Q of 2 ps. The fourth case was the ideal case, and we expected to have better results compared to the other cases.

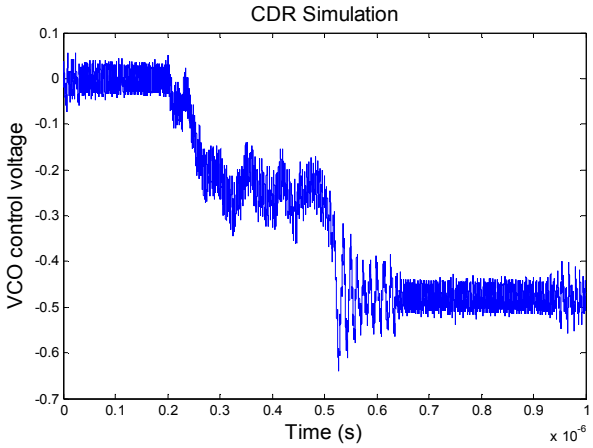


Figure 3.9: CDR simulation result for case 1

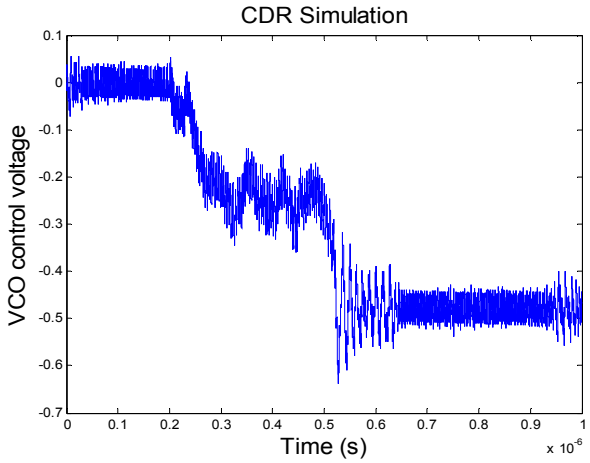


Figure 3.10: CDR simulation result for case 2

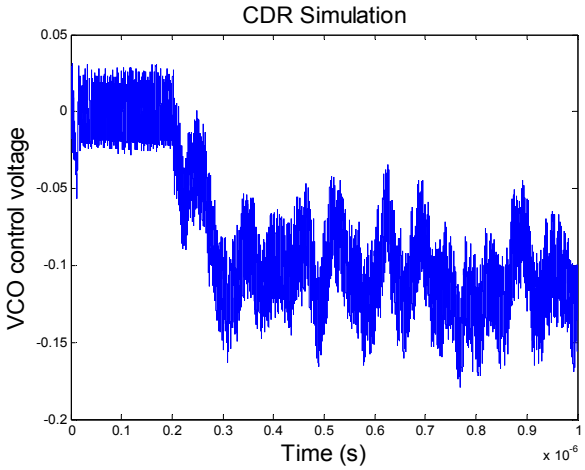


Figure 3.11: CDR simulation result for case 3

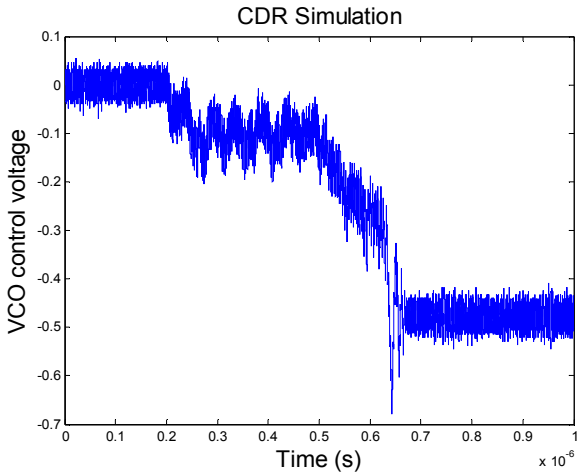


Figure 3.12: CDR simulation result for case 4

Table 3. CDR information for various timing metrics

Case	T_{su} [ps]	T_{ho} [ps]	T_{cq} [ps]	10 G Lock Time [ns]	9.5 G Lock Time [ns]	10 G PP Jitter [UI]	9.5 G PP Jitter [UI]
1	10	10	20	≈10	≈450	≈0.063	≈0.086
2	1	19	20	≈10	≈450	≈0.091	≈0.087
3	19	1	20	≈20	NL	≈0.067	—
4	1	1	2	≈1	≈450	≈0.061	≈0.084

In case 1, setup and hold times were equal, and C2Q time was two times greater than each of them. Prior to 0.2 μs, the CDR was locked at approximately 10ns. When the switch selected PRBS7 9.5 Gbps, the control voltage was affected by the ripples, and the CDR got locked at approximately 450 ns, as shown in Figure 3.9. The VCO control voltage experienced a peak-to-peak jitter of 0.063 UI for PRBS7 10 Gbps and 0.086 UI for PRBS7 9.5 Gbps. Cases 2 and 3 are the best demonstrations of timing parameter effects on the system performance. In case 2, lower setup time and higher hold time made the model to lock at approximately 10 ns for PRBS7 10 Gbps and 450 ns for PRBS7 9.5 Gbps. It means that choosing lower setup time and higher hold time will force the CDR to show a behavior similar to the previous case from lock time perspective (the VCO control voltage experienced a peak-to-peak jitter of 0.091 UI for PRBS7 10 Gbps data generator and 0.087 UI for PRBS7 9.5 Gbps). In case 3, hold time was higher than setup time, and as shown in Figure 3.11, the CDR never got locked or in other words, took a long time to get locked, which is highly undesirable. The goal of the current modeling was to diminish the lock time and also jitter components. So, case 3 was the worst case from lock time and jitter aspects. Case 4 introduced the ideal case, and therefore, better results compared to the previous cases were expected. The simulation shown in Figure 3.12 and also results illustrated in Table 3 proved the accuracy of the prediction. The CDR became locked considerably fast, and the VCO control voltage contained lower jitter compared to the other cases (0.061 UI for PRBS7 10 Gbps and 0.084 UI for PRBS7 9.5 Gbps). We can conclude that the best selection for the optimum result is to have an equal setup and hold times. This conclusion is quite reasonable. With

equal setup and hold times, jitter margin becomes maximum, and the BER is reduced significantly.

3.4. Bang-Bang PD Characteristics

Table 4 shows bang-bang PD characteristic information for different setup and hold summations. Figures 3.13, 3.14, 3.15, and 3.16 demonstrate the characteristics for various summations. In Figure 3.13, setup and hold summation was equal to 20 ps. Three cases were observed for a total summation of 20 ps. In cases 1 and 3 with unequal setup and hold times, dead zone width was equal to 18 ps. In case 2 with equal setup and hold times, the dead zone width was decreased by 2 ps. In this simulation, equal setup and hold times led to a smaller dead zone width, which is highly desirable. Figure 3.14 shows the phase detector characteristic for setup and hold summation of 10 ps. The dead zone width was decreased compared to the previous simulation. Therefore, lower setup and hold summation will lead to a lower dead zone width. In cases 1 and 3 with unequal setup and hold times, the dead zone width is lower than the one with equal setup and hold times. The result is totally different from what we observed in the previous simulation.

Figure 3.15 shows the impact of lower setup and hold summation on the dead zone width. In this examination, the summation was chosen to be 6 ps. The dead zone width was equal to 4 ps for all cases. Two observations were achieved from this simulation: First, the lower the summation is, the lower the dead zone width is. Second, by choosing lower setup and hold summation, equality or inequality of setup and hold times will not affect the dead zone width.

Table 4. Bang-bang PD characteristic information

Case	Fig. 3.12	T_{su} [ps]	T_{ho} [ps]	T_{cq} [ps]	Dead Zone Width	Case	Fig. 3.13	T_{su} [ps]	T_{ho} [ps]	T_{cq} [ps]	Dead Zone Width
1	——	5	15	20	18	1	——	2	8	20	6
2	— ▪ — ▪	10	10	20	16	2	— ▪ — ▪	5	5	20	8
3	▪▪▪▪▪	15	5	20	18	3	▪▪▪▪▪	8	2	20	6
Case	**Fig. 3.14**	**T_{su} [ps]**	**T_{ho} [ps]**	**T_{cq} [ps]**	**Dead Zone Width**	**Case**	**Fig. 3.15**	**T_{su} [ps]**	**T_{ho} [ps]**	**T_{cq} [ps]**	**Dead Zone Width**
1	——	1	5	10	4	1	——	10	20	20	26
2	— ▪ — ▪	3	3	10	4	2	— ▪ — ▪	15	15	20	No Char.
3	▪▪▪▪▪	5	1	10	4	3	▪▪▪▪▪	20	10	20	No Char.

Higher setup and hold summations degraded the CDR performance considerably. Figure 3.16 shows the result for a setup and hold summation of 30 ps. In case 1, the setup time was lower than the hold time, and the related dead zone width was considerably high. In cases 2 and 3, the typical bang-bang characteristics were vanished and replaced with straight lines. The results indicated that by increasing the summation, the phase detector would not operate as a bang-bang system anymore, leading to significant errors at the output.

By comparing the results achieved in this section and the previous section, we realized that there was a perfect compatibility between the results. Both cases implied that lower setup and hold times would generate results that are close to the ideal case.

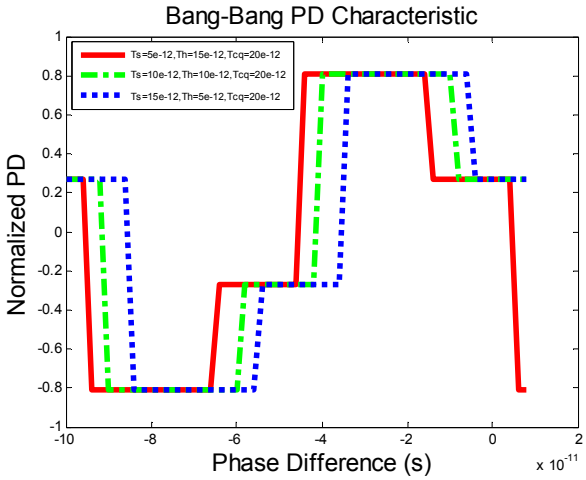


Figure 3.13: PD characteristic with setup and hold summation equal to 20 ps

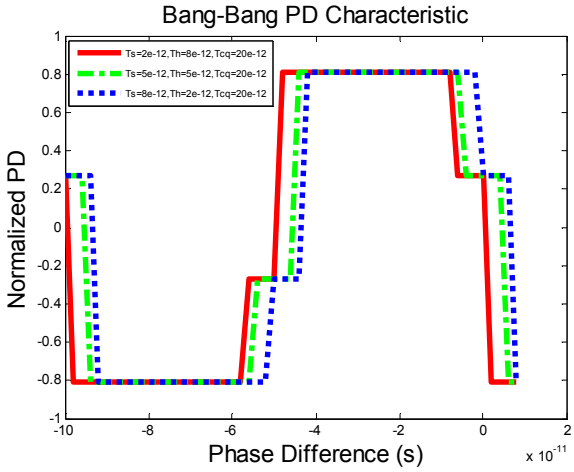


Figure 3.14: PD characteristic with setup and hold summation equal to 10 ps

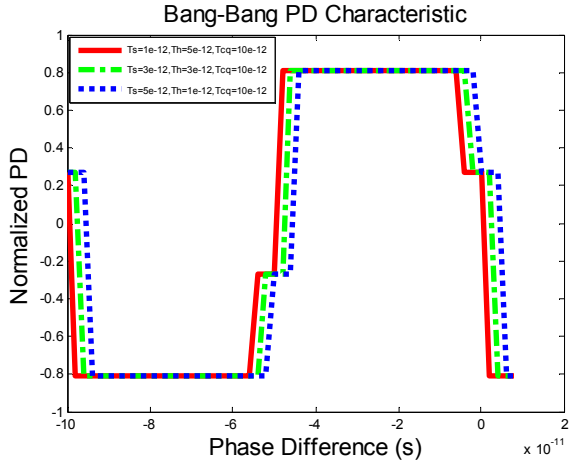


Figure 3.15: PD characteristic with setup and hold summation equal to 6 ps

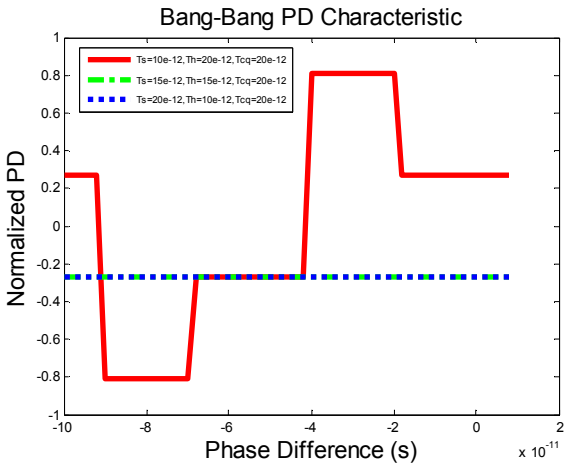


Figure 3.16: PD characteristic with setup and hold summation equal to 30 ps

3.5. PD with Various DFF Timing Parameter Values

In previous simulations, four DFFs in the phase detector structure were allocated similar setup, hold, and C2Q times. Now, we want to study the behavior of the system when different values are allocated to DFF timing parameters and observe the variations in lock time and peak-to-peak jitter.

Table 5 shows the results for various cases, which were selected randomly. Case 1 manifested high jitter and relatively long lock time. So, this result was far away from the study goal. Cases 2 and 6 did not generate a locked behavior at the VCO control voltage, and consequently, no peak-to-peak jitter could be measured for these cases. Cases 3-5 show relatively lower numbers compared to the other cases. Yet, these results could not compete with the ones generated in Table 3.

Table 5. CDR with Alexander PD information for various timing metric values

Case	DFF1			DFF2			DFF3			DFF4			PP Jitter [ps]	PRBS9.5 Lock Time [ns]
	SU [ps]	HO [ps]	C2Q [ps]	SU [ps]	HO [ps]	C2Q [ps]	SU [ps]	HO [ps]	C2Q [ps]	SU [ps]	HO [ps]	C2Q [ps]		
1	1	19	40	10	10	40	10	10	40	10	10	40	9.8	≈750
2	19	1	40	10	10	40	10	10	40	10	10	40	—	NL
3	10	10	40	10	10	40	10	10	40	1	19	40	9.5	≈700
4	10	10	40	19	1	40	10	10	40	10	10	40	9.6	≈700
5	1	19	40	1	19	40	10	10	40	10	10	40	9.5	≈800
6	19	1	40	19	1	40	10	10	40	10	10	40	—	NL

3.6. DFF Calibration Technique

As discussed in the previous section, the best case was achieved with equal setup and hold times. Due to process variations, the timing metrics do not remain constant. The goal is to adjust timing parameters (i.e., setup and hold times) in a way that they become equal. Obviously, this is a very challenging task, especially when the data are provided by PRBS sources. The idea is to utilize delay blocks to achieve the goal. We started with finding DFF setup and hold times inside an Alexander PD. To do that, two counters were used in the path of the clock. Data for the counters were provided by a 0101 data source. Note that the counters are enabled for a limited time, which would be sufficient to determine setup and hold times. First, we tried the proposed model for equal setup and hold times to observe how much delay time we needed for other cases. With equal setup and hold times, we expected to see no change in the locking point. After

having the desired value for the delay time, we could adjust any setup and hold times to hit midpoint of the data eye. Figure 3.17 illustrates the DFF calibration technique qualitatively. It shows how the clock rising edge was adjusted to sample the midpoint of the unit interval. This task was performed for cases with higher setup time and lower hold time or lower setup time and higher hold time. After the DFF calibration was conducted, it was expected to have a behavior close to case 1. Due to random value generation at metastability points, we had different lock points. Note that this difference is ignorable.

Figure 3.18 shows the DFF calibration model. The generated clock signal at the output of the CDR was fed to the setup and hold time counters to measure the precise timing metrics. After monitoring setup and hold times, the amount of clock sampling edge delay was calculated by the timing adjuster block. The result was applied to a delay block to shift the clock. Note that a switch was provided in the clock path to redirect the delayed signal to the clock port of the CDR. The switch let the delayed clock pass through whenever setup and hold time measurements were concluded.

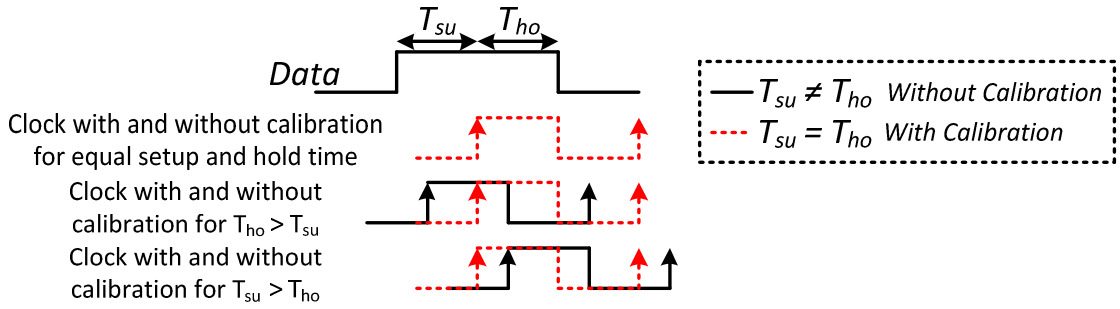


Figure 3.17: Clock rising edge adjustment

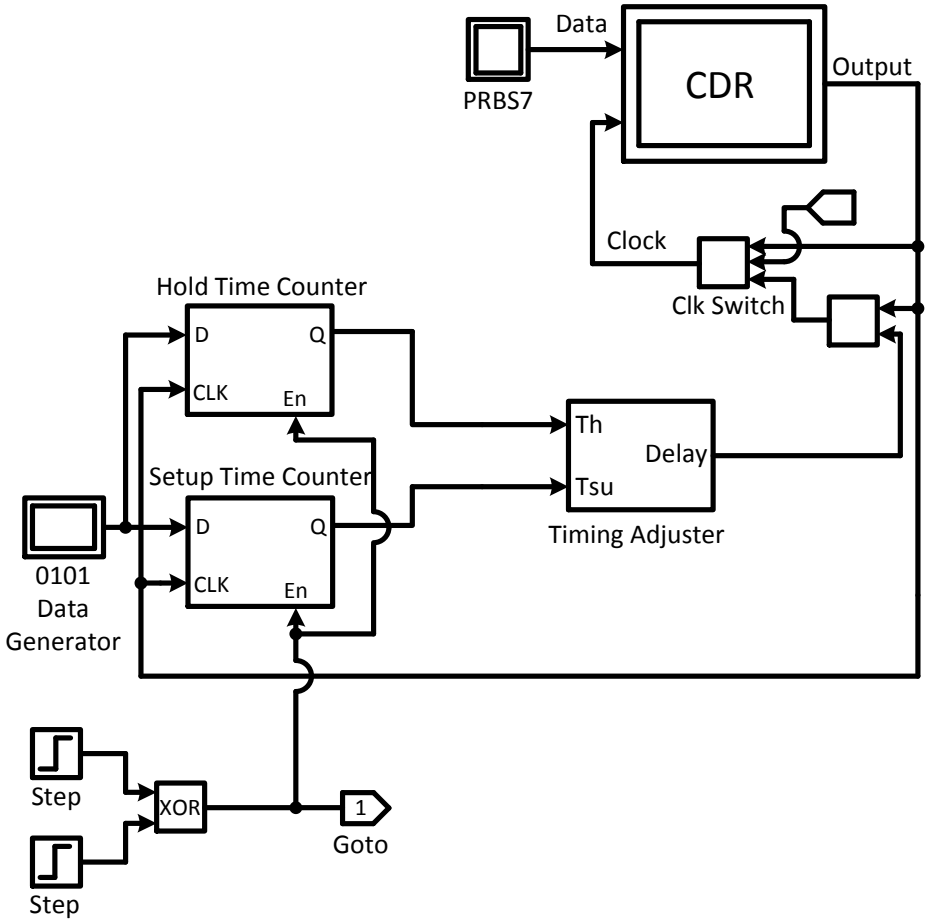


Figure 3.18: DFF calibration model

Figures 3.19, 3.20, 3.21, and 3.22 depict the results before and after calibration for various timing metrics. Simulations show four different cases before and after calibration. Note that for T_{su} = 5 ps and T_{ho} = 15 ps and vice versa, no lock occurred before calibration. However, the CDR became locked around the lock point of equal setup and hold times after calibration.

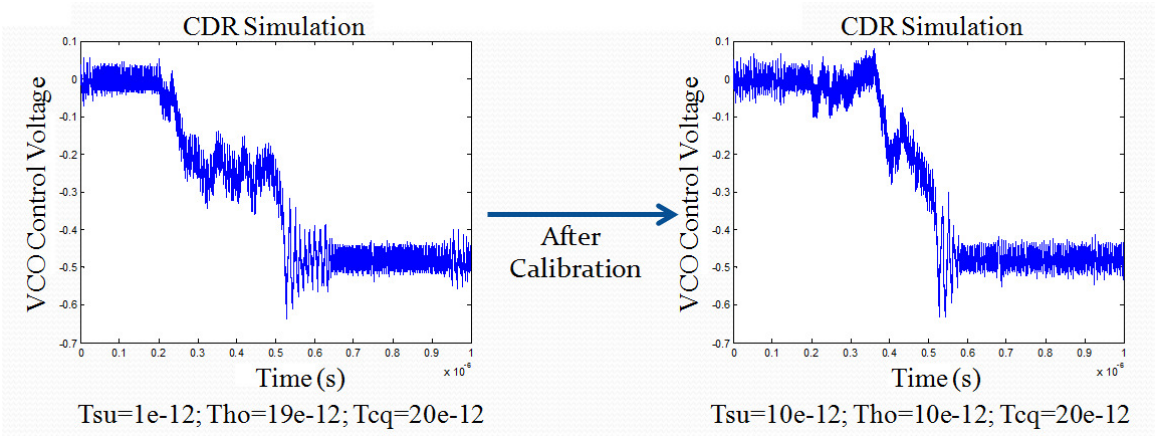


Figure 3.19: Calibration simulation result for T_{su} = 1 ps and T_{ho} = 19 ps

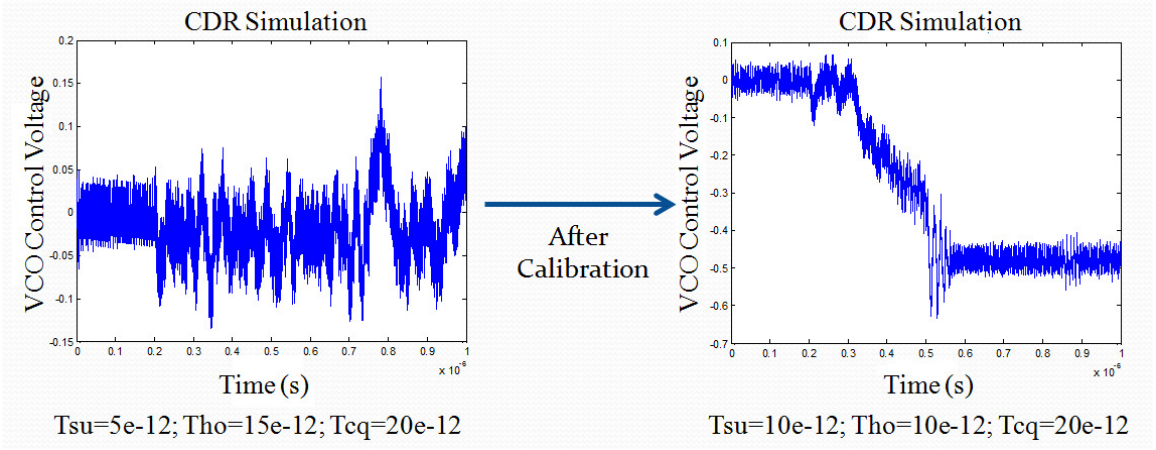


Figure 3.20: Calibration simulation result for T_{su} = 5 ps and T_{ho} = 15 ps

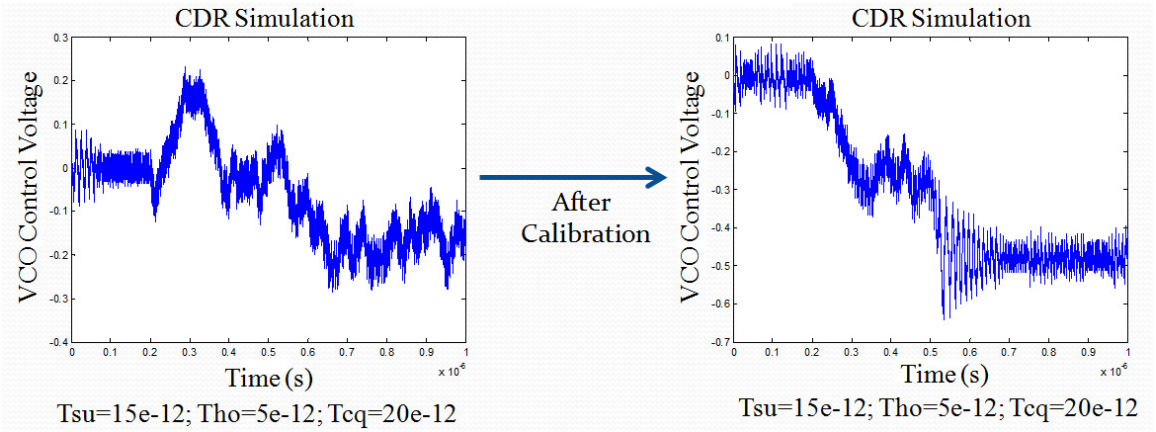


Figure 3.21: Calibration simulation result for T_{su} = 15 ps and T_{ho} = 5 ps

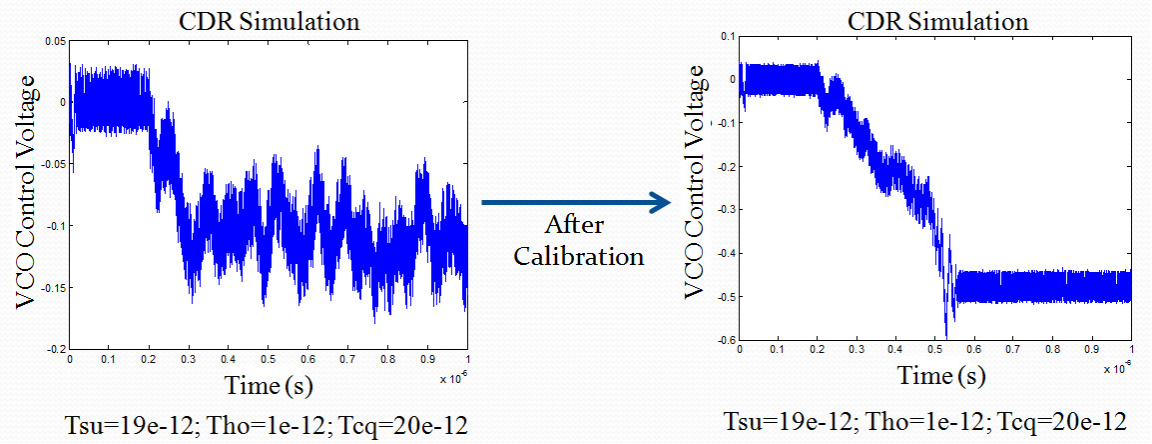


Figure 3.22: Calibration simulation result for T_{su} = 19 ps and T_{ho} = 1 ps

3.7. Phase Detector Architectures

This section discusses the modeling of various linear and binary phase detectors and their impacts on the CDR performance. Each PD architecture incorporated the modeled DFF metastability. Eventually modeled linear and binary PDs were utilized in the CDR model and simulated. The CDR lock time and jitter components were monitored and compared to determine the optimum result.

3.7.1. Hogge PD

A single DFF can be utilized to perform phase detection. The drawback of using a DFF as PD is the high amount of jitter generation. In the absence of data transitions, the charge pump injects current into the loop filter, causing jump in the VCO control line [4]. Eventually a high amount of jitter will be resulted. To alleviate the drawback, the input and output of the DFF were XORed. The resulting signal is called "proportional pulses." The current structure suffered from another drawback, called "data pattern

dependency." For two different data patterns, the phase difference between input and output could be the same. To remedy this flaw, second DFF was added next to the first one, and the outputs of first and second DFFs were XORed to generate "reference pulses." These signals remove any phase ambiguity [4]. Figure 3.23 depicts Hogge PD block diagram. Note that DFF1 and DFF2 become activated with rising and falling edges of the clock respectively. Figure 3.24 illustrates input and output signals. The modeled DFF placed in Hogge PD architecture, and the resulting PD placed in the CDR model of Figure 3.7. The input data were provided by a pseudo random bit sequence (PRBS) generator. For 0.3 µs, PRBS7 10 Gbps generated the incoming data. At 0.3 µs, PRBS7 9.5 Gbps took over and generated random data stream. Two different data sources were utilized to observe the system reaction in different situations. Figures 3.25 and 3.26 show the VCO control voltage and the phase detector linear characteristic respectively. The CDR lock time for PRBS7 10 Gbps and PRBS7 9.5 Gbps were 10 ns and 0.9 µs respectively. Figure 3.25 implies that Hogge PD is a linear phase detector. It means the average output voltage of the PD varies linearly with respect to the phase difference between the input data and the generated clock signal. Note that because the modeled DFF manifested nonideal behaviors, the characteristic contained some nonlinearities.

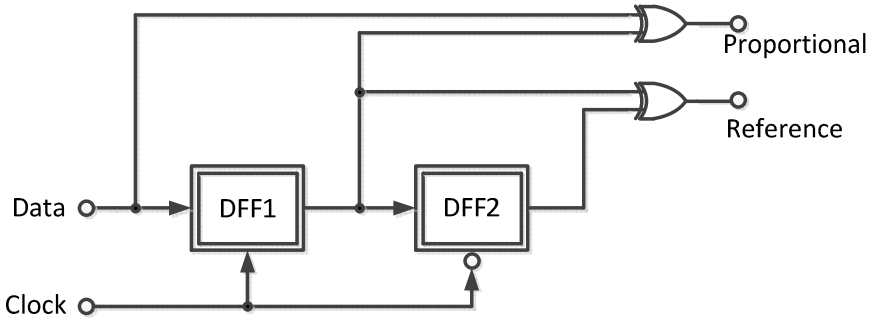


Figure 3.23: Hogge PD model

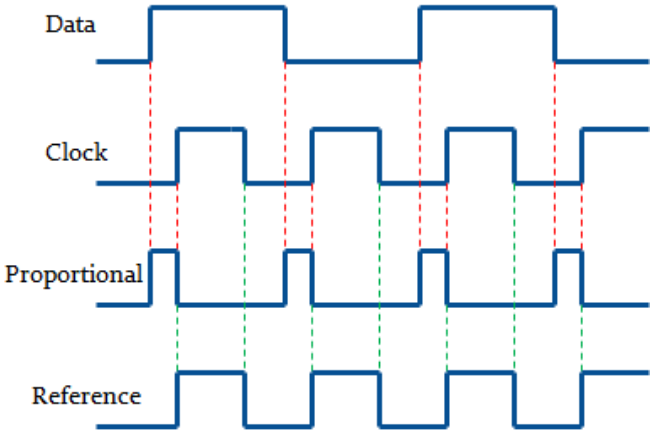


Figure 3.24: Hogge PD input and output signals

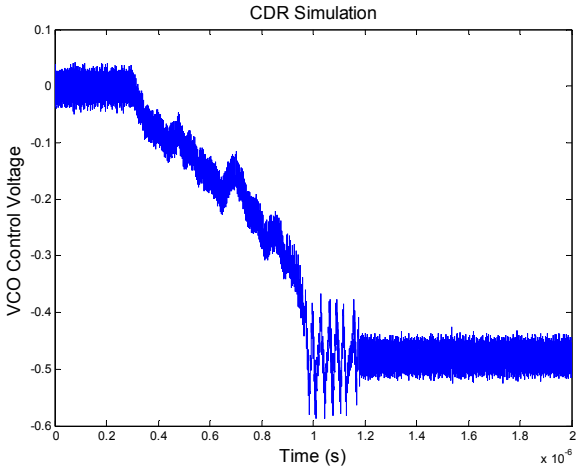


Figure 3.25: VCO control voltage for Hogge PD

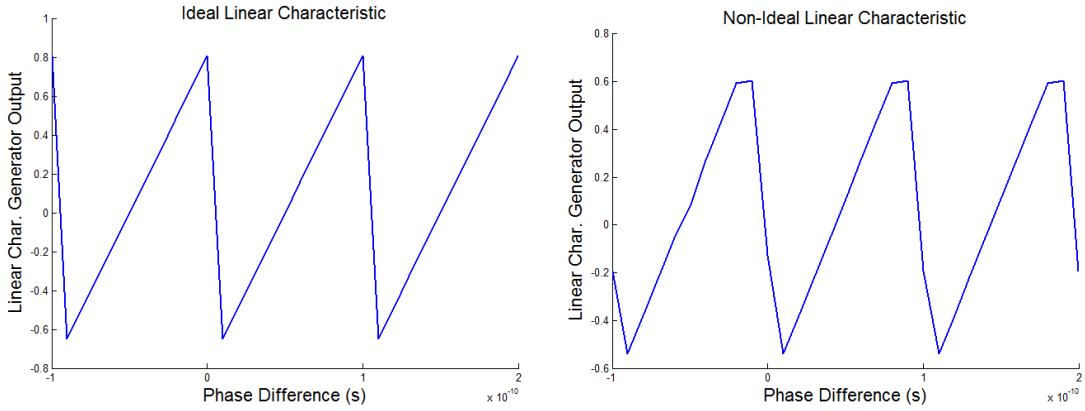


Figure 3.26: Ideal and nonideal linear characteristics for Hogge PD

3.7.2. Alexander PD

Figure 3.27 demonstrates Alexander PD block diagram. In this method, three samples of the data, shown in Figure 3.28, were extracted by three consecutive clock edges, and edge and phase difference detections were performed based on those samples. If S1 and S2 are on the same data level, and S3 is on the opposite level, then the clock is early. If S2 and S3 are on the same data level, and S1 is on the opposite level, then the clock is late. Any other case is translated to no data transition. Alexander PD is also called "tri-state PD." It means the PD has only three states during operation [2] [11].

By rising edges of the clock, DFF1 samples the incoming data, and the result is delayed by one clock cycle by DFF2. Falling edges of the clock applied to DFF3 sample the data, and the results are delayed by half a clock cycle by DFF4. Note that the only task of DFF2 and DFF4 is to delay the sampled data. Some Alexander PD designs tend to replace DFF2 and DFF4 with a digital block that contains Verilog codes. Running Verilog codes can generate the required delay, alleviating the need for hardware.

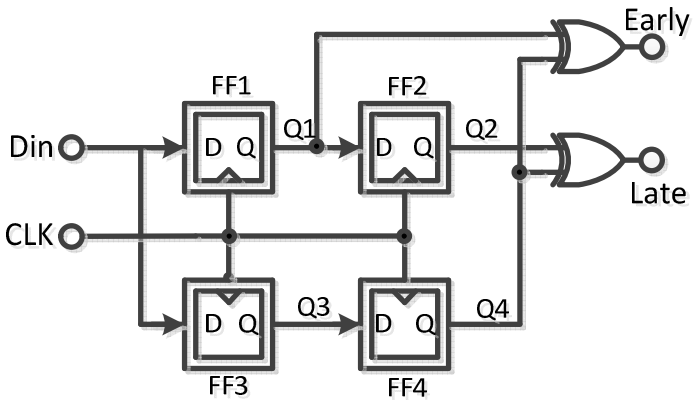


Figure 3.27: Alexander PD model

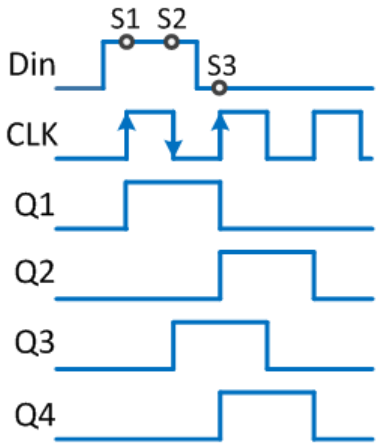


Figure 3.28: Three consecutive clock edges

When an Alexander PD is placed in the CDR model, the locking condition is achieved if second samples extracted by DFF3 appear in the vicinity of data transition points. In this case, FF3 and FF4 will be in the metastability region. The simulation results, shown in sections 3.3 and 3.4, clarify the impact of this architecture on the CDR performance. The results imply that Alexander PD is a bang-bang system (i.e., when the phase difference becomes zero, the average output jumps from one extreme to another).

3.7.3. Half-Rate PD

Linear phase detectors are used in cases where low jitter on the control line is needed. For high-speed purposes, linear PDs do not react well. Therefore, bang-bang phase detectors are utilized. Designing an oscillator with proper tuning range and acceptable jitter is a challenging task. Because of that, engineers use half-rate phase detectors. In such PDs, the clock frequency is half the input data rate. If, for example, the input data rate is 10 Gbps, the clock frequency will be 5 GHz. The half-rate PDs

proposed by Savoj [3] and Joram [13] are examples of linear PDs. Proportional and reference pulses are generated by using four latches and two XOR gates. This approach, due to less charge pump activity, yields lower jitter on the control line. However, for high-speed applications, linear half-rate methodology manifests relatively weak performance. Hence, the early-late method, which is highly compatible with high-speed environments, is discussed in this section.

Figure 3.29 depicts a bang-bang half-rate PD block diagram. The clock sampling edges are shown in Figure 3.30. Random data were applied to each DFF, and four clock edges with a phase difference of 90 degrees sampled the input data [14]. Two consecutive sampling results were XORed and fed to the voltage-to-current converter block. Eventually the V/I block outputs were summed and sent to the loop filter for generating the control voltage for the VCO. Because the clock frequency is decreased, the phase detector and the frequency divider (if present) will relax from speed aspect. By placing the PD inside the CDR model, the VCO control voltage and the bang-bang characteristics, shown in Figures 3.31 and 3.32, were resulted. The lock time for PRBS7 10 Gbps and PRBS7 9.5 Gbps data generators were approximately 5 ns and 500 ns respectively. Note that for all types of PDs simulated in this section, the setup and hold times were equal to 10 ps, and C2Q time was equal to 20 ps. Two types of bang-bang characteristics were generated for the half-rate PD. In the ideal case, ideal DFFs, which do not show the metastability behavior, were utilized. As shown in the figure, a clear bang-bang characteristic was achieved. However, in the nonideal case, which DFF metastability was placed in the PD structure, a highly nonlinear characteristic was

obtained. As mentioned before, by reducing transistor timing metrics, the system performance converges to the results obtained in the ideal case. The simulation results showed that for setup and hold times less than or equal to 5 ps, the nonideal bang-bang characteristic resembled the ideal counterpart.

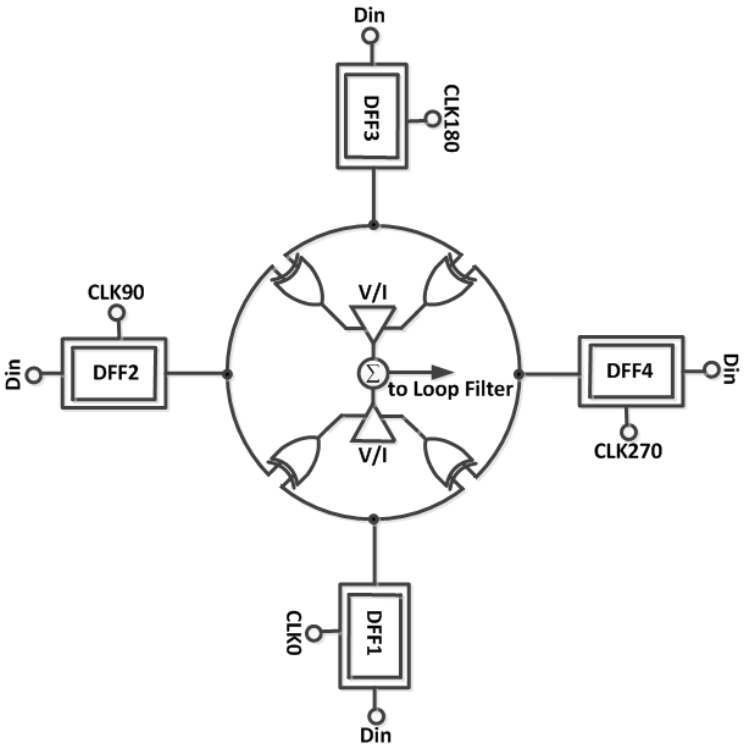


Figure 3.29: Bang-bang half-rate PD block diagram

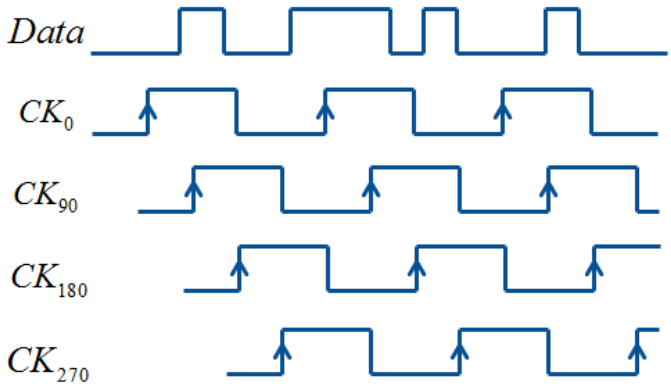


Figure 3.30: Half-rate PD sampling edges

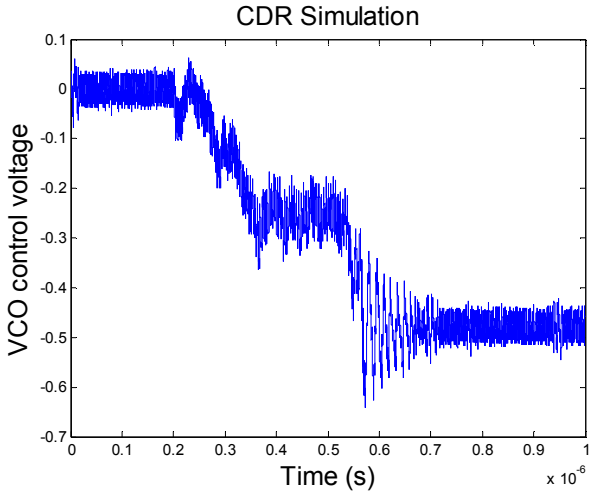


Figure 3.31: VCO control voltage for half-rate PD

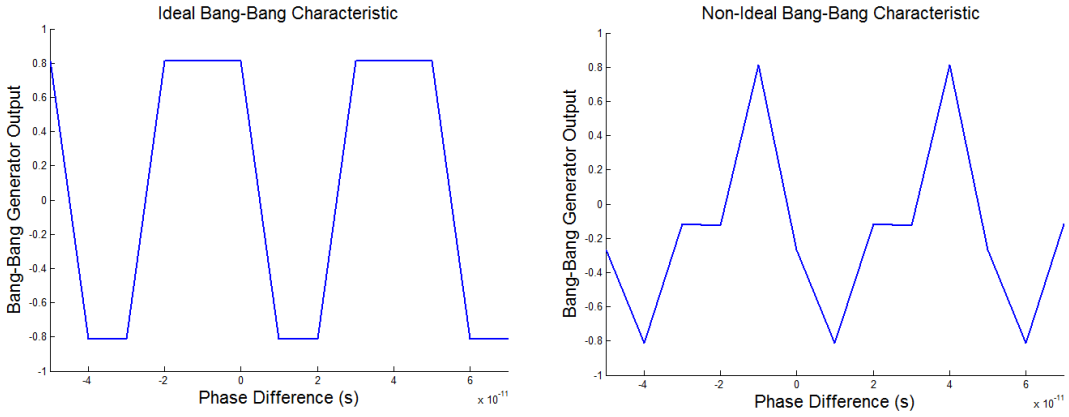


Figure 3.32: Ideal and nonideal bang-bang characteristics for half-rate PD

3.7.4. Quarter-Rate PD

In quarter-rate PDs, the clock frequency is $\frac{1}{4}$ of the incoming data rate. With the data rate of 10 Gbps at the input, the clock frequency is 2.5 GHz. For linear PDs, both rising and falling edges of in-phase and quadrature-phase of the clock are utilized to sample the data. CLK_0, CLK_{90} , CLK_{180} , and CLK_{270} are the clock phases sampling the

data stream. Designs proposed by Saffari [15], Tontisirin [16], and Alavi [17] demonstrate linear quarter-rate phase detectors. However, for a bang-bang system, eight phases of the clock are used to sample the input data. The clock phases are CLK_0, CLK_{45}, CLK_{90} ... CLK_{315}. Figure 3.33 shows a bang-bang $\frac{1}{4}$-rate PD. This architecture was derived from Alexander phase detection logic. Every two consecutive samples were evaluated by boolean XOR function to determine any data transition. Note that in the absence of data transitions, all samples remain constant. Hence, high and low levels will not be generated, and eventually we expect to have low jitter on the control line. The clock phases were generated by using delay gates. The PD outputs were applied to four charge pumps. As mentioned before, charge pump subtracts two inputs, and the result is multiplied by the current required to be injected into the loop filter. Figure 3.34 illustrates the clock phases. Eight data samples are extracted in each clock cycle. Four of them are monitored in half a clock cycle. Therefore, the discussed system is called "quarter-rate PD." The PD proposed by Lee [18] demonstrates a bang-bang tri-state quarter-rate PD. Figure 3.35 depicts the result for the VCO control voltage.

According to Figure 3.35, the quarter-rate PD got locked at approximately 10 ns for PRBS7 10 Gbps and 800 ns for PRBS7 9.5 Gbps. Setup and hold times were equal to 10 ps for both cases. From jitter aspect, the CDR contained relatively low values. For PRBS7 10 Gbps, the control line experienced a peak-to-peak jitter of 0.11 UI, and for PRBS7 9.5 Gbps, the peak-to-peak jitter was equal to 0.15 UI. Therefore, most sophisticated designs tend to utilize half and quarter-rate PDs in CDR architectures. It is interesting to know that by increasing the number of devices, the power dissipation is

comparable to the full-rate Alexander PD [4]. For instance, in quarter-rate phase detection, eight DFFs conduct phase detection, data retiming, and demultiplexing. Nonetheless, in full-rate PDs, six DFFs and one latch perform phase detection and data demultiplexing. Therefore, the power dissipation remains relatively constant by replacing PD in the CDR architecture [18]. From clock load capacitance perspective, full-rate Alexander PDs provide larger capacitance compared to the quarter-rate bang-bang PDs [18].

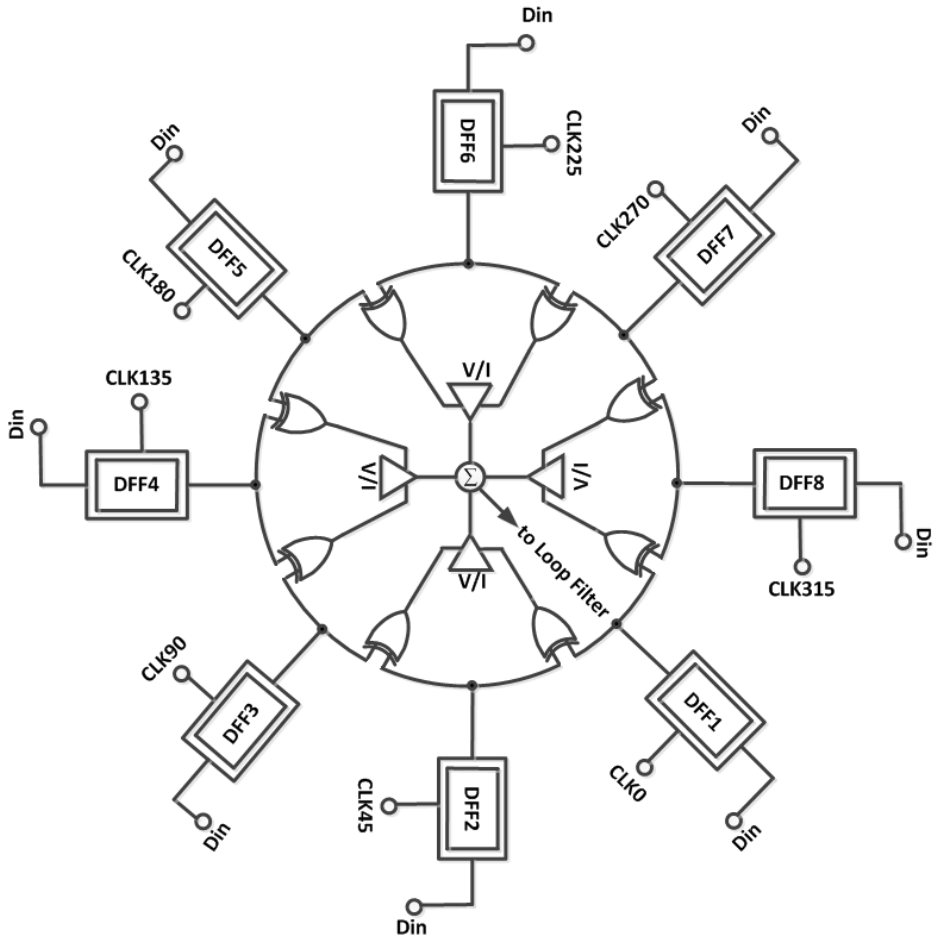


Figure 3.33: Bang-bang quarter-rate PD model

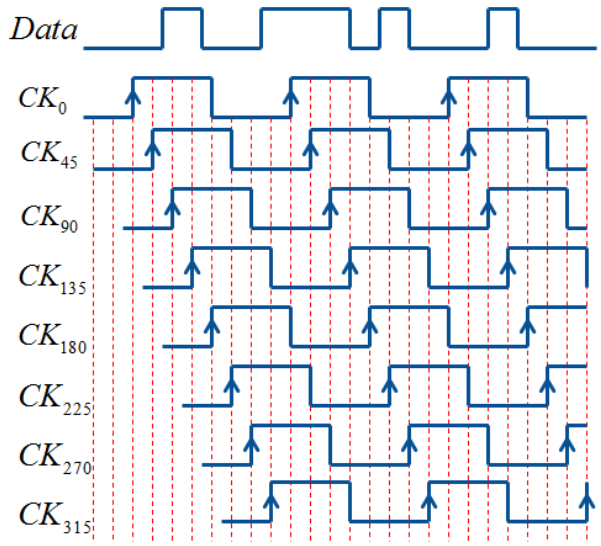


Figure. 3.34: Clock sampling edges for quarter-rate PD

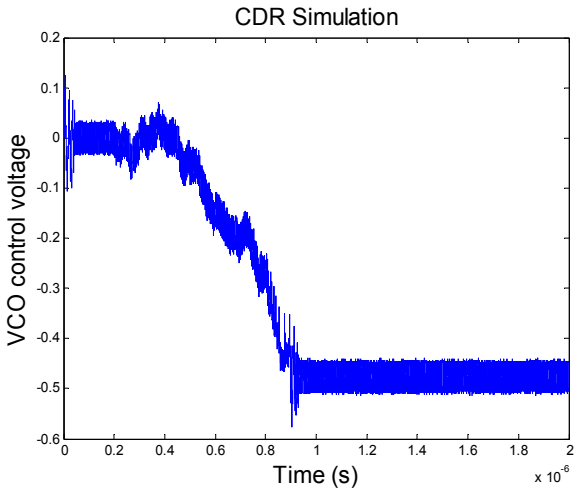


Figure 3.35: VCO control voltage for quarter-rate PD

Ideal and nonideal bang-bang characteristics were generated and shown in Figure 3.36. In the ideal case, a very clear bang-bang figure was observed without a dead zone region. Nevertheless, in nonideal case with metastability DFFs inside the architecture, the dead zone and metastability factors were monitored. Note that by reducing setup and

hold times, the metastability impact was alleviated. Eventually we confronted a system that manifested more ideal behaviors.

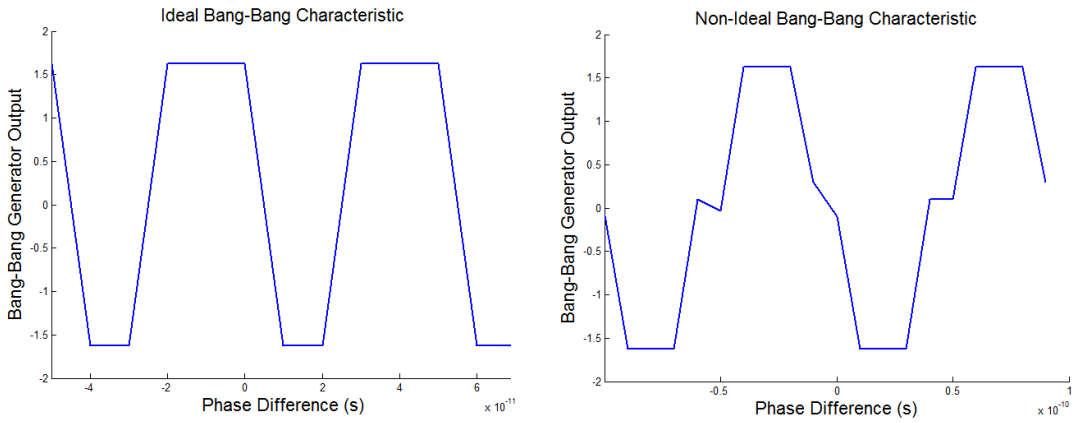


Figure 3.36: Ideal and nonideal bang-bang characteristics for quarter-rate PD

3.7.5. Octant-Rate PD

Another option for very high-speed purposes is octant-rate PD. The designs proposed by Song [19] and Seo [20] manifest linear 1/8-rate PDs. In this section, a bang-bang octant-rate PD, which follows the Alexander phase detection method, is introduced. To model a bang-bang PD, sixteen DFFs and XORs were utilized. Similar to Alexander and quarter-rate PDs, every two consecutive samples were compared to detect data transitions. In such PDs, the incoming data are sampled by sixteen clock phases: CLK_0, $CLK_{22.5}$, CLK_{45} ... $CLK_{337.5}$. The input data are sampled sixteen times per clock cycle. In half a clock cycle, eight data samples are extracted. Figure 3.37 shows the bang-bang octant-rate PD block diagram. The Alexander phase detection methodology was utilized for the PD modeling. In other words, the quarter-rate bang-bang PD was expanded to

perform phase detection with higher accuracy. The sixteen clock sampling edges are shown in Figure 3.38. Because each half of the clock cycle is sampled for eight times, this structure is also called "octant or 1/8-rate phase detector." The main drawback of this methodology is the high number of elements utilized in the model. The number of consisting components were doubled compared to the previous case (quarter-rate). Hence, the area occupied by the PD is considerably large.

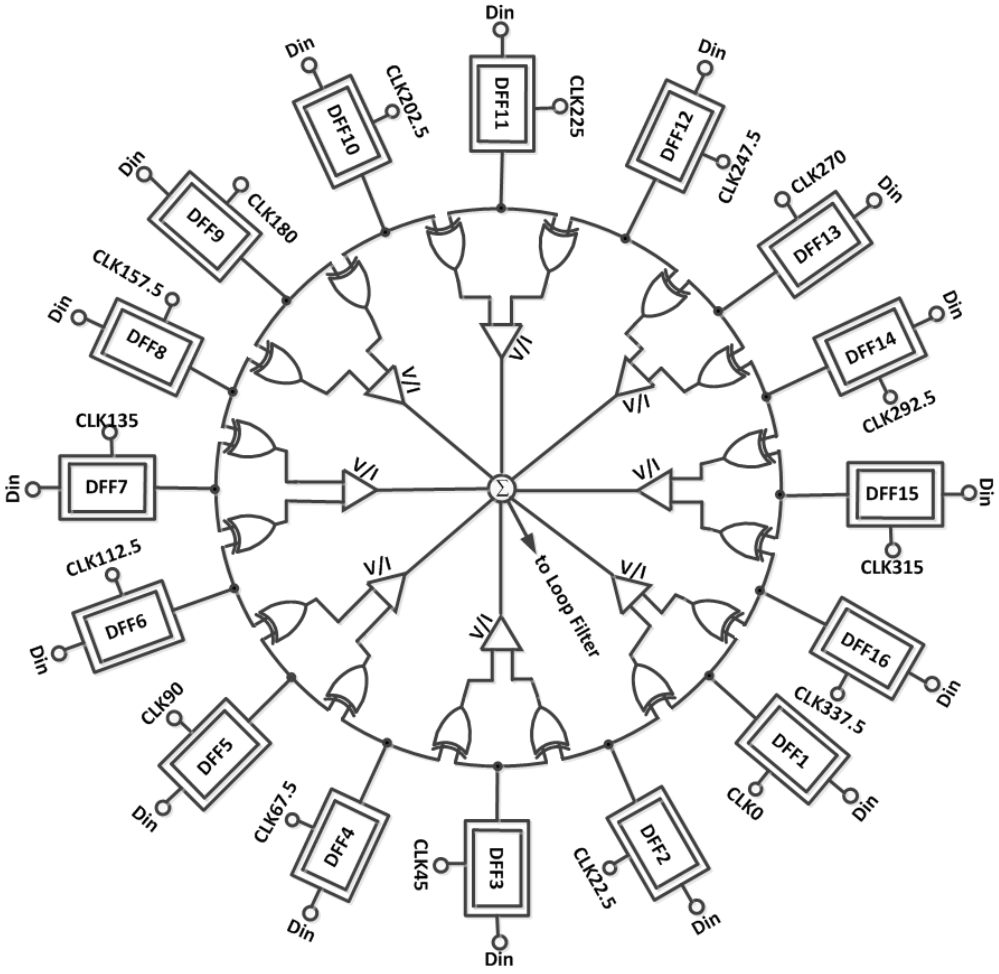


Figure 3.37: Bang-bang octant-rate PD model

The generated clock by the VCO was delayed by a fixed value to generate the clock phases. The value depends on the input data frequency. Note that the value should be chosen properly. This architecture, similar to quarter-rate PDs, suffers from having high numbers of components. Sixteen DFFs and XORs will occupy a large area, and this issue is highly undesirable, especially in transistor-level designs. Current designs tend to scale down the area occupied by circuit components. Moreover, utilizing a high number of transistors leads to a significant amount of delays and substantial speed reduction.

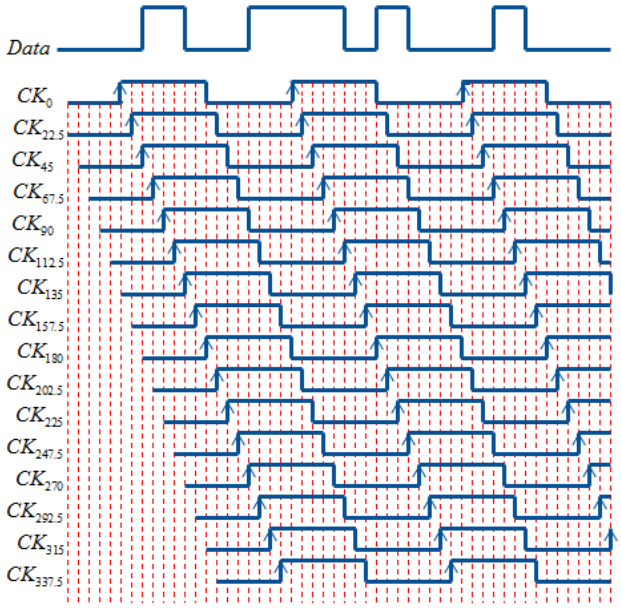


Figure 3.38: Clock sampling edges for octant-rate PD

Figures 3.39 and 3.40 illustrate the results for the CDR, which incorporates 1/8-rate bang-bang PD. As shown in the result, the CDR got locked at 20 ns for PRBS7 10 Gbps and at approximately 1600 ns for PRBS7 9.5 Gbps, which is considerably lower value compared to half-rate and quarter-rate PDs. Peak-to-peak jitter for PRBS7 10 Gbps

and PRBS7 9.5 Gbps was 0.14 UI and 0.21 UI respectively. PRBS7 10 Gbps generated high jitter on the control line due to high charge pump activity. To conclude, this method manifests long lock times and a high amount of jitter on the control line. Nevertheless, the low clock frequencies, provided by linear and binary octant-rate PDs, make this approach attractive in some specific applications.

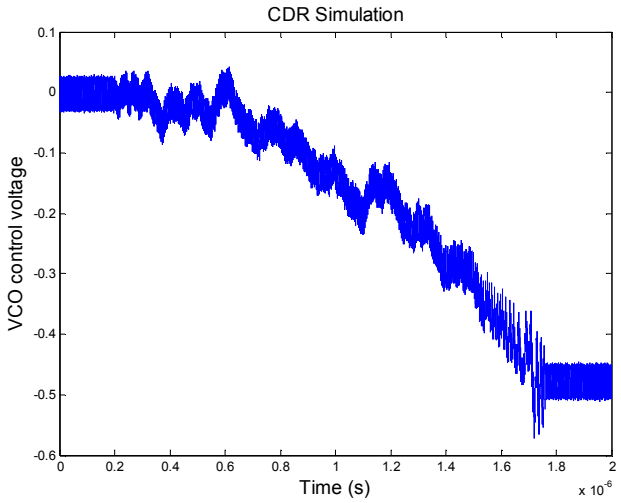


Figure 3.39: VCO control voltage for octant-rate PD

Bang-bang characteristics were generated for ideal and nonideal cases and shown in Figure 3.40. The ideal case manifested a very clear bang-bang graph. In the nonideal case, the dead zone factor was apparent. By reducing timing parameters, the dead zone width shrank, and the effect of metastability vanished. Table 6 summarizes the lock time and peak-to-peak jitter for the PDs discussed in this chapter. Note that the peak-to-peak jitter values were calculated based on the information extracted from the eye diagrams. Figure 3.41 depicts the eye diagram for the fourth case of Alexander PD. The quality of CDR circuits is judged based on the parameters shown in the eye diagram [21] [22].

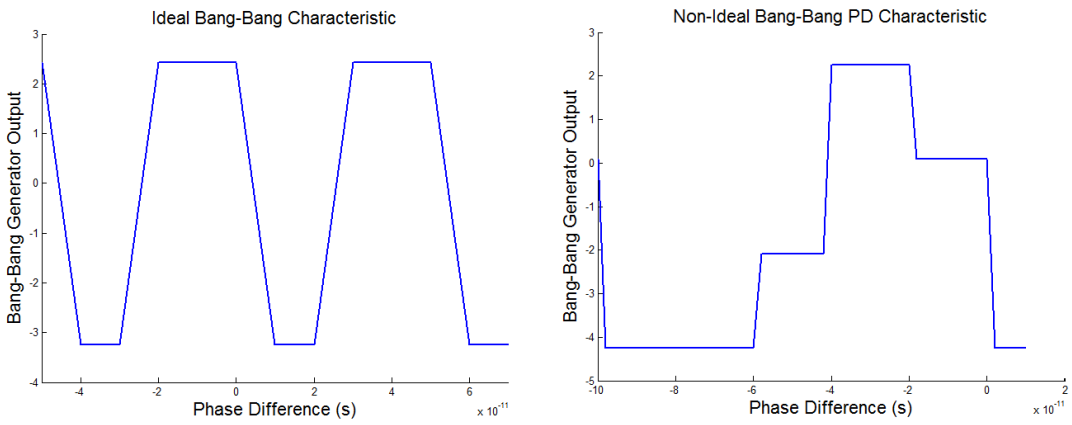


Figure 3.40: Ideal and nonideal bang-bang characteristics for octant-rate PD

Table 6. Simulation results for various phase detectors

PD	PD Type	Timing Metrics			PRBS7 10Gbps Lock Time [ns]	PRBS7 9.5Gbps Lock Time [ns]	P-P Jitter 10Gbps [UI]	P-P Jitter 9.5Gbps [UI]
		T_{su} [ps]	T_{ho} [ps]	T_{cq} [ps]				
Hogge	Linear	10	10	20	≈10	≈900	≈0.01	≈0.04
Alexander	Binary	1	19	20	≈10	≈450	≈0.091	≈0.087
		10	10	20	≈10	≈450	≈0.063	≈0.086
		19	1	20	≈20	NL	≈0.067	—
		1	1	2	≈1	≈450	≈0.061	≈0.084
Half-Rate	Binary	10	10	20	≈5	≈500	≈0.07	≈0.1
Quarter-Rate	Binary	10	10	20	≈10	≈800	≈0.11	≈0.15
Octant-Rate	Binary	10	10	20	≈20	≈1600	≈0.14	≈0.21

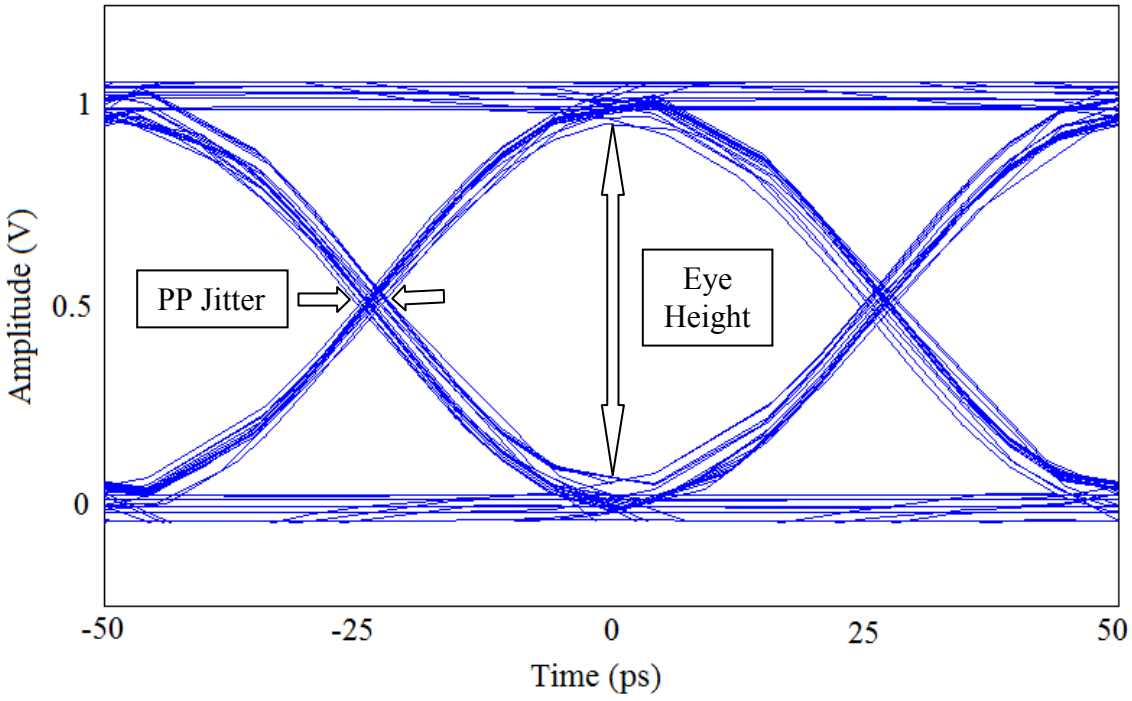


Figure 3.41: Eye diagram for fourth case of Alexander PD

Chapter 4. CDR Design

4.1. Verilog-A Modeling

In this section, using Cadence Virtuoso, Verilog-A codes were generated for each of the CDR building blocks, and finally the simulation results were obtained. To implement CDR in Cadence Virtuoso, the model proposed in Figure 3.7 was utilized. To provide a data stream for the phase detector, PRBS and switch blocks were used. To switch between two data generators, a multiplexer block was provided by Verilog-A codes. The CDR was simulated for 3 μs, and the selector port of the multiplexer was triggered at 1 μs to change the random sequence. The PRBS blocks were constructed by using random data generator commands. For phase detection, first, DFFs were modeled ideally (i.e., metastability was ignored), and the CDR simulation results were extracted. Then, the metastability factor was considered, and the results were generated based on different timing factors. The charge pump output was calculated by subtraction and multiplication of the inputs (i.e., up and down signals at the output of phase detector). Analog loop filter was utilized to filter out the output of the previous stage and generate the control voltage for the next stage. A resistor (R=1 KΩ) and a capacitor (C1=4 pF) were provided in series, and then a second capacitor (C2=400 fF) was placed in parallel to prevent large jumps (ripples) on the control voltage. The VCO block was built by considering Equation 2.3. The free-running frequency and K_{vco} were equal to 9.5 GHz and 1 GHz/V respectively. The data rates were set to 10 Gbps and 10.2 Gbps (i.e., 10

Gbps for 1 μs and 10.2 Gbps for 2 μs). It is important to mention that a block was employed at the output of the data stream generator and the VCO to make sure that the signal makes a transition from low data level to high data level and vice versa. Figure 4.1 depicts the CDR model in Cadence Virtuoso using Verilog-A blocks.

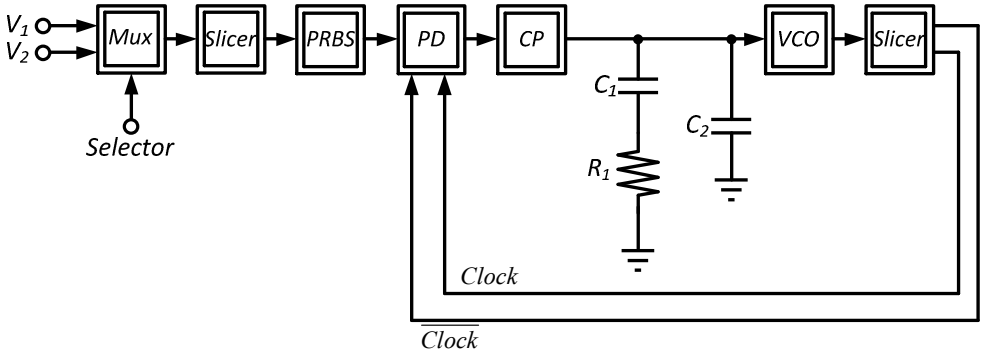


Figure 4.1: CDR model using Verilog-A blocks

4.1.1. Multiplexer (Mux), Slicer, and Random Data Generator

In Figure 4.1, switch and relay blocks were replaced by multiplexer and slicer blocks. At 1 μs, the step pulse switched the periodic input pulse propagated to the output of the multiplexer. The block was programmed based on this methodology. The slicer block limits the signal levels, applied to the PRBS, to zero and one. When the input level of the slicer becomes greater than 0.5, the output gets high. Otherwise, the output makes a transition to the low level (zero). The PRBS block can be programmed by using the shift register logic. The idea is to construct a shift register with enough length, so that after a long time the pattern repeats. Another approach for PRBS implementation is to utilize DFF and XOR blocks generated by Verilog-A codes. Verilog-A based DFF and

XOR blocks are discussed in the next section. Figure 4.2 shows the PRBS generator output.

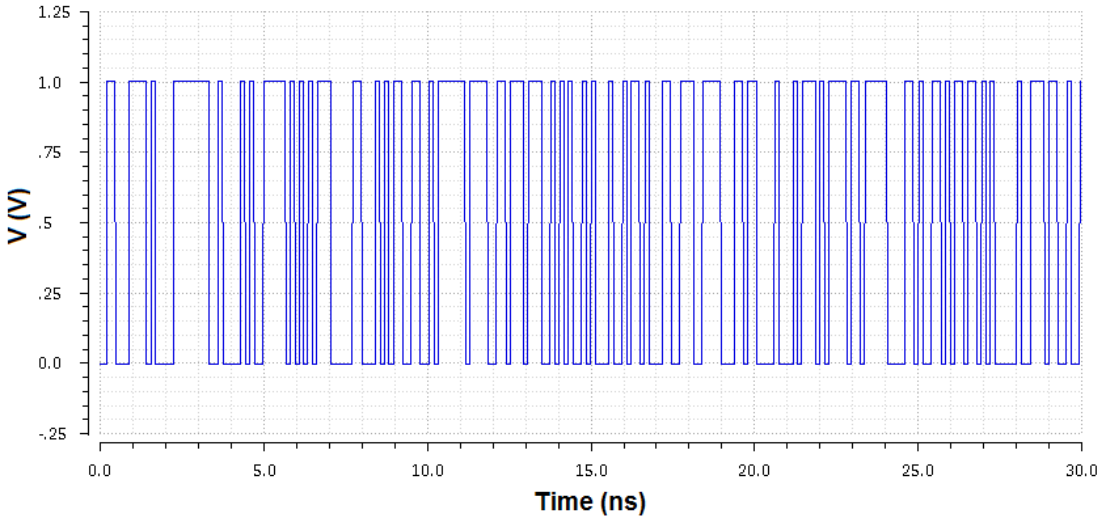


Figure 4.2: Verilog-A based PRBS generator output

4.1.2. Ideal and Nonideal DFFs

Figures 4.3 and 4.4 illustrate the results for without and with violation DFFs respectively. The first trace is the DFF output. The second and third ones are the clock and the input data. The data and clock periods, setup, hold, and C2Q times were 10 ps, 5 ps, 1 ps, 1 ps, and 2 ps respectively. In Figure 4.3, the clock delay was set to 2 ps. Therefore, by considering the data period, setup, and hold values, we can conclude that no violation occurred in this case. As shown in Figure 4.3, the DFF output is a replica of the input with 2 ps delay, which is the propagation delay time. In the second case, the clock delay was set to 0.9 ps. Therefore because the setup time requirement was not fulfilled, a random data at the output of the DFF were generated (Figure 4.4). The

random output will also be obtained if the hold time violation occurs. The ideal DFF did not show metastability behavior, and it produced the output without propagation delay. Also, setup and hold violations were ignored. Note that all Verilog-A codes are provided in section A of Appendix.

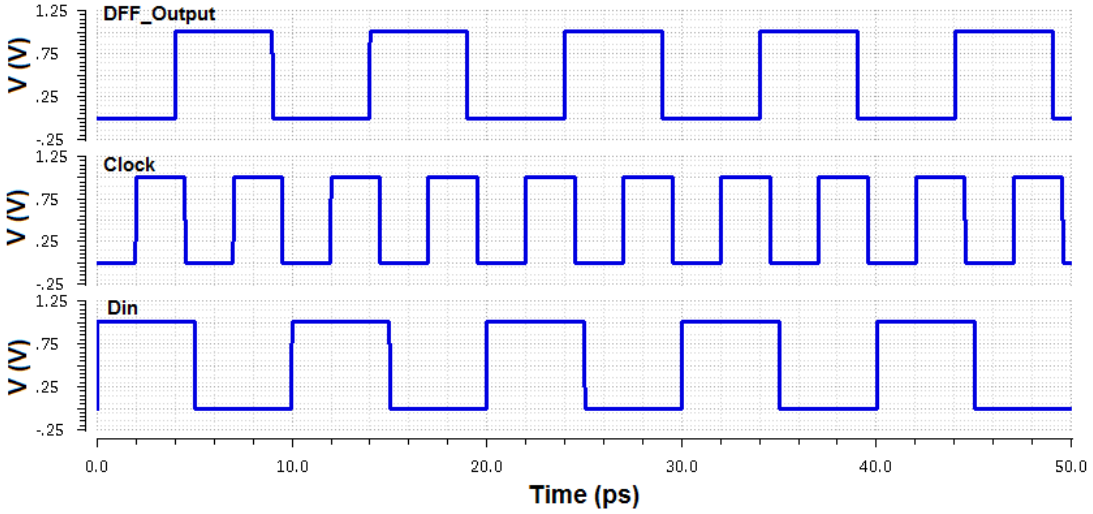


Figure 4.3: Verilog-A DFF simulation result without violation

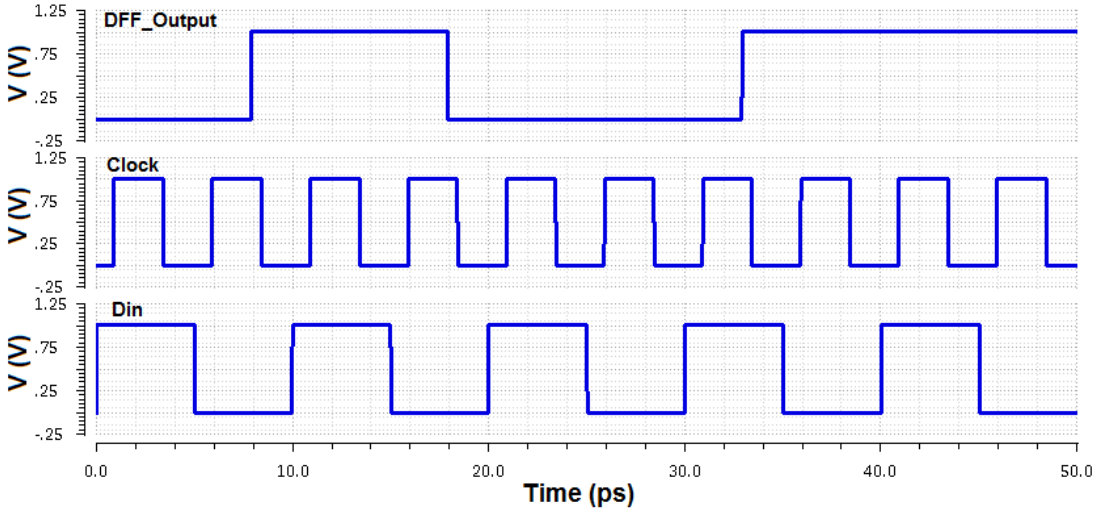


Figure 4.4: Verilog-A DFF simulation result with violation

4.1.3. Exclusive-OR (XOR), CP, and VCO

Two-input XORs were utilized in PRBS and phase detector blocks. To model XOR in Verilog-A environment, two internal parameters were defined and devoted to the inputs. For instance, parameter "a" was allocated to the first input and parameter "b" to the second one. If XOR inputs become greater than or equal to 0.5, the related parameter becomes high. Otherwise, it becomes low. Eventually the summation of two defined parameters determines the XOR output. The code is provided in Appendix. Subtraction and multiplication are two main tasks of CP blocks and can be easily created by Verilog codes. Eventually the VCO block was constructed by considering Equation 2.3. Figure 4.5 illustrates up, down, input random sequence, the CP output, and the VCO output respectively. Up and down signals are the Alexander PD outputs. The last trace manifests a sinusoidal waveform with variable frequencies.

4.1.4. CDR Simulation Results

Figures 4.6, 4.7, 4.8, and 4.9 depict the VCO control voltage for various cases. In Figure 4.6, ideal DFFs were utilized in the phase detector architecture. As mentioned before, the data rate within 1 μs was set to 10 Gbps, and the free-running frequency was 9.5 GHz. Hence, the CDR should be locked at 500 mV. At 1 μs, when the data rate was switched to 10.2 Gbps, the CDR got locked at 700 mV. The result confirmed hand calculations. Figure 4.7 demonstrates the control voltage for case 1, provided in Table 7. The setup and hold times were equal, and the summation was equal to 20 ps. The result, shown in Figure 4.7, verified the discussion. Figure 4.8 shows the control voltage for case

2. For 10 Gbps data rate, no lock occurred. For 10.2 Gbps, the CDR got locked fast; however, the amount of ripple on the control voltage was relatively high. In Figure 4.9, setup and hold times were equal, and the summation was equal to 2 ps. For both data generators, the CDR became locked, and the amount of ripple on the control voltage was relatively small. This case was close to the ideal case. In case 3, with higher setup time and lower hold time, the VCO control voltage was similar to the second case (i.e., Figure 4.9). So, we conclude that equal setup and hold times with small summations give us the best results from lock time and jitter aspects. Similar results were achieved in the Simulink experiments.

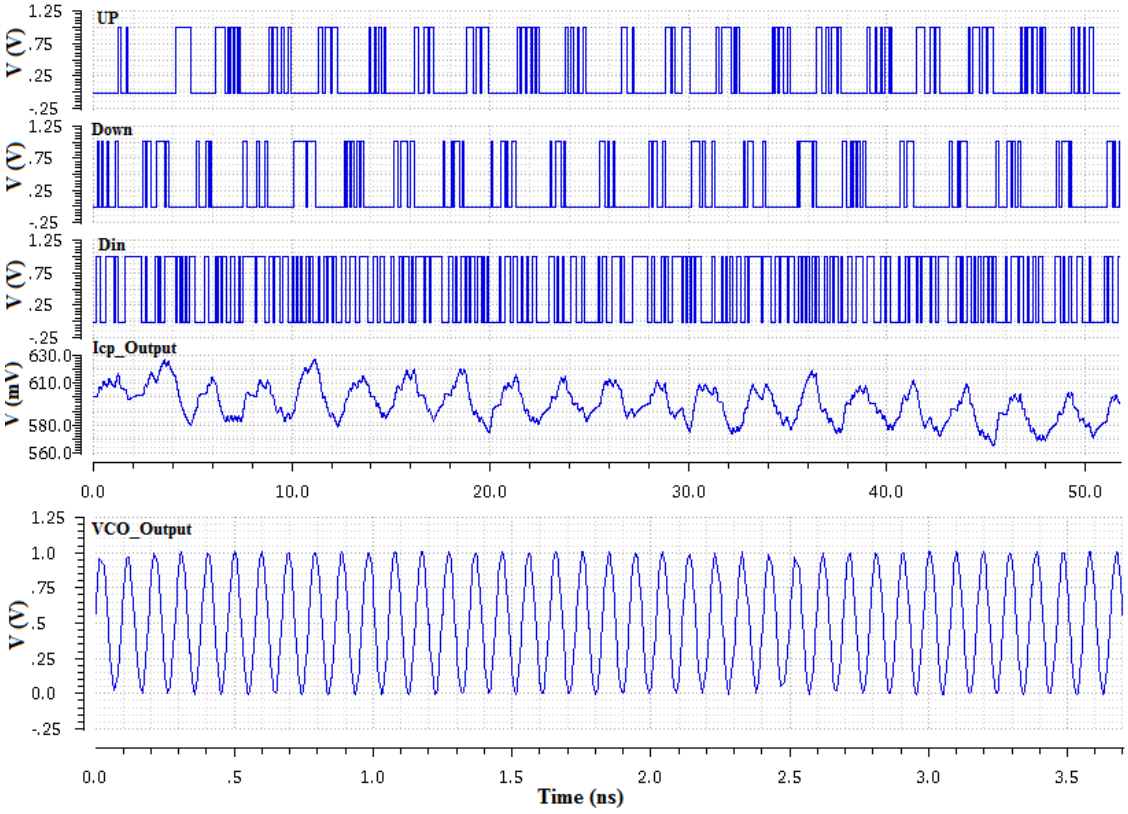


Figure 4.5: Signals at various points of the CDR loop

The information provided in Table 7 compares different cases from lock time and peak-to-peak jitter perspectives. As expected, the ideal case showed lower lock time compared to the other cases. Lock behavior was observed for both data generators. Furthermore, significantly low values were obtained for peak-to-peak jitter. With equal and small setup and hold times, relatively similar results were achieved. The lock time and peak-to-peak jitter values were slightly different.

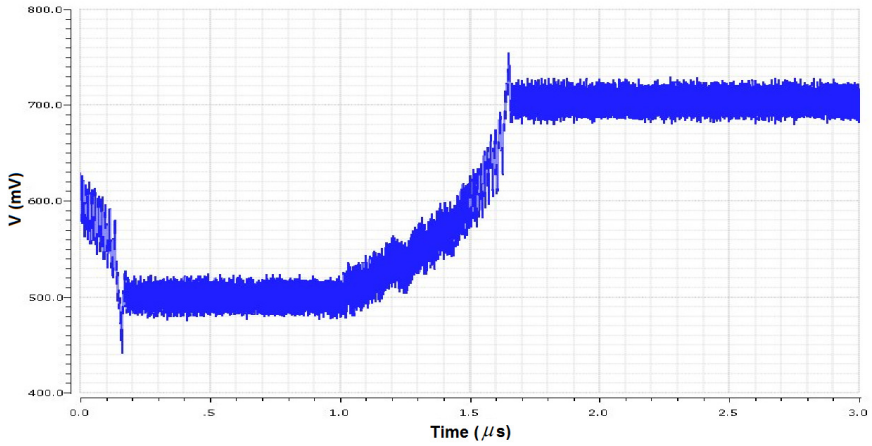


Figure 4.6: VCO control voltage for ideal DFF

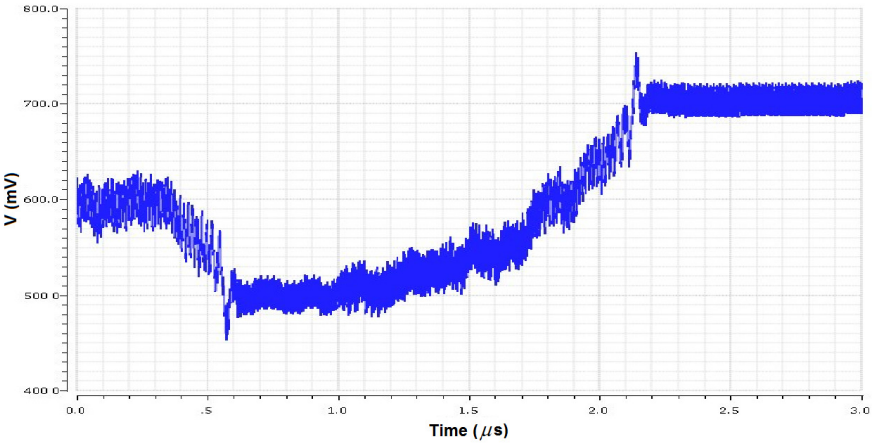


Figure 4.7: VCO control voltage for case 1

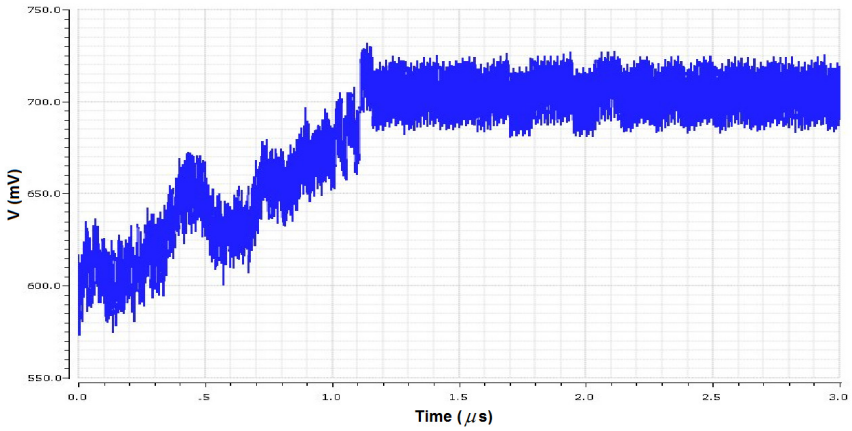


Figure 4.8: VCO control voltage for case 2

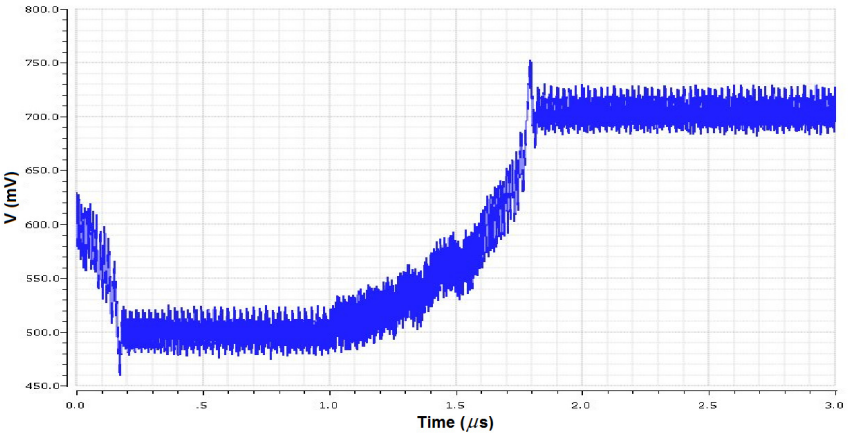


Figure 4.9: VCO control voltage for case 4

Table 7. CDR information for various timing metrics

Case	**Tsu [ps]**	**Tho [ps]**	**Tcq [ps]**	**10 G Lock Time [ns]**	**10.2 G Lock Time [ns]**	**10 G PP Jitter [UI]**	**10.2 G PP Jitter [UI]**
1	10	10	20	≈600	≈1200	≈0.058	≈0.093
2	1	19	20	NL	≈200	NL	≈0.2
3	19	1	20	NL	≈200	NL	≈0.2
4	1	1	2	≈200	≈800	≈0.043	≈0.062
Ideal	-	-	-	≈200	≈600	≈0.025	≈0.029

4.1.5. Metastability and Bang-Bang Characteristics

In this section, bang-bang characteristics were generated for various timing cases. In the previous section, the impacts of metastability and timing metrics on lock time and peak-to-peak jitter were studied. We expect the same effect in the characteristics. It means that as long as the setup and hold times are close to the ideal case, clear bang-bang characteristics are resulted. However, when timing metrics get far from the ideal point, dead zone and metastability factors will disturb the performance severely. Figures 4.10, 4.11, 4.12, 4.13, and 4.14 demonstrate the bang-bang characteristics of the ideal, fourth, second, first, and third cases respectively. In the ideal case, a clear characteristic was observed without any dead zone region. No metastability factor was included in the coded DFF, and the result was quite logical. The outcome was completely similar to the one generated in Simulink environment. In the fourth case with equal setup and hold times and close to the ideality, a characteristic similar to the previous case was extracted. Note that some nonlinearities were observed and pointed out in Figure 4.11.

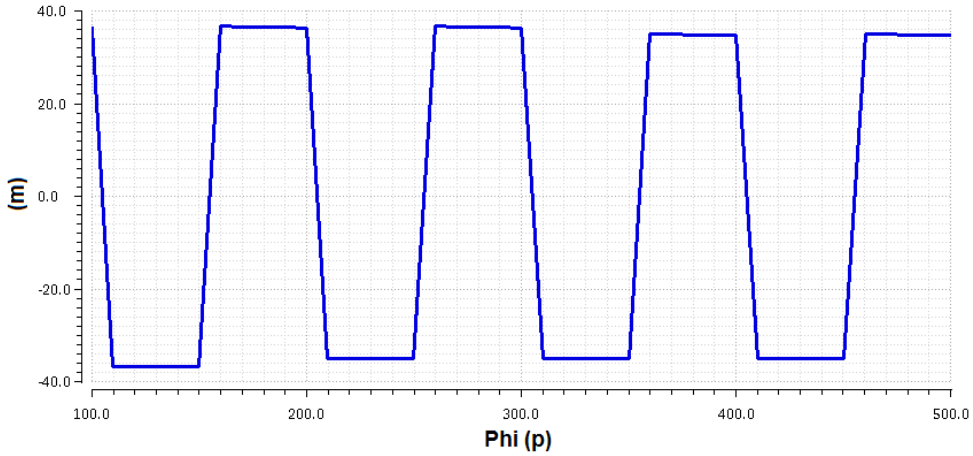


Figure 4.10: Ideal case bang-bang characteristic

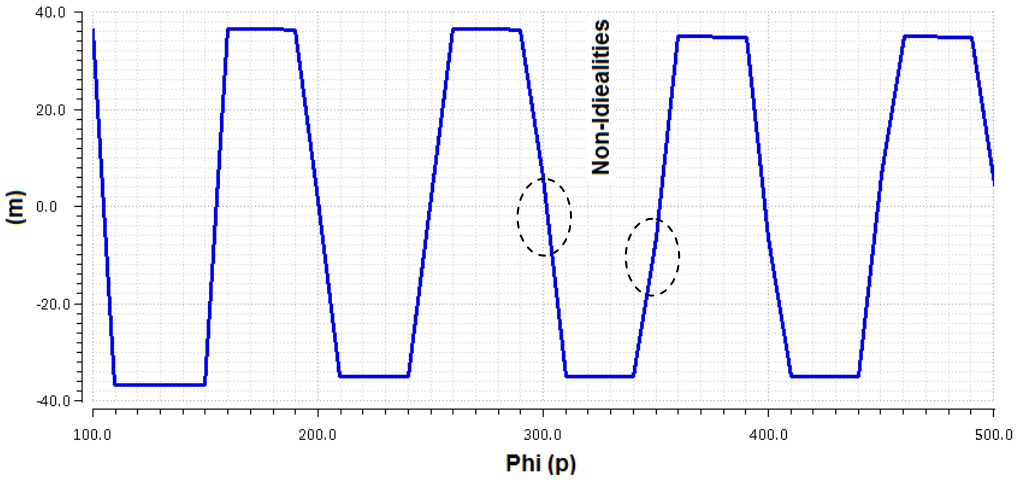


Figure 4.11: Bang-bang characteristic for case 4

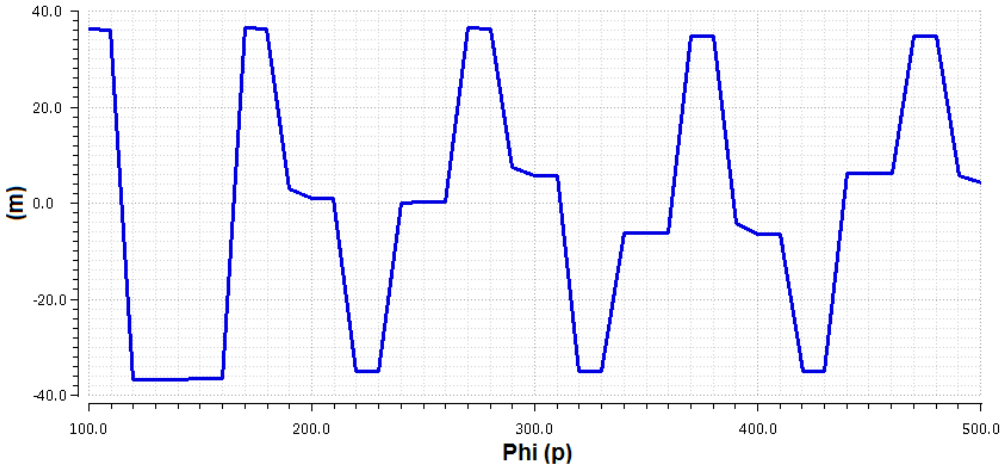


Figure 4.12: Bang-bang characteristic for case 2

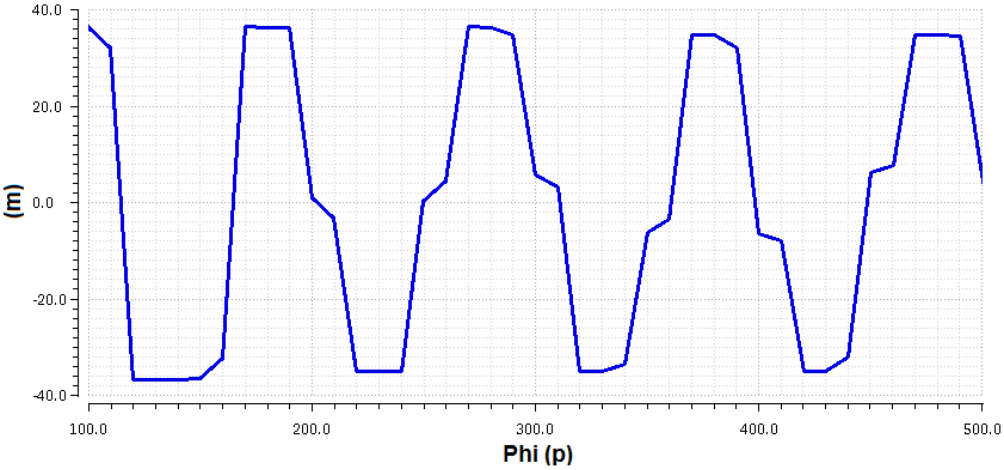


Figure 4.13: Bang-bang characteristic for case 1

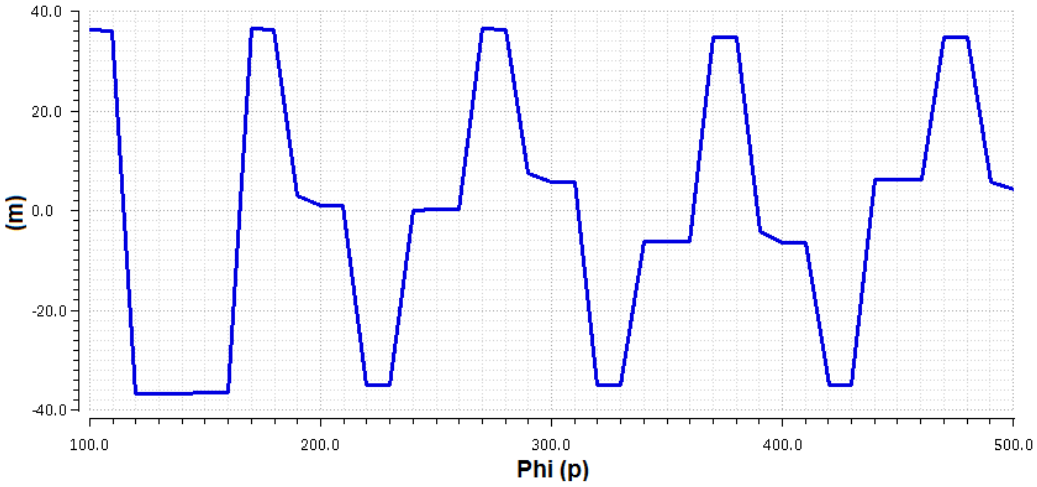


Figure 4.14: Bang-bang characteristic for case 3

4.1.6. Verilog-A Based Bang-Bang Phase Detectors

In the previous chapter, half-rate, quarter-rate, and octant-rate phase detectors were modeled and simulated. In this section, PDs were built by Verilog-A codes, and the results were extracted and compared to the ones in the previous chapter.

The phase detector shown in Figure 3.28 was translated into Verilog-A environment. The DFF metastability with setup and hold times equal to 10 ps was used in the PD architecture. In the Simulink model, the clock edges were delayed by one-fourth of the clock cycle. Delay blocks with the external delay control signal adjusted the clock edges. In Cadence Virtuoso, the delay function was performed by one line code, which is provided in Appendix. At the output of the PD, a 2-bit adder was placed. By replacing the Alexander PD with the modeled one, the VCO control voltage shown in Figure 4.15 was resulted. The amount of jitter on the control line was considerably low. If we extend the methodology performed in the half-rate phase detection, lower-rate phase detectors are

produced. The architectures introduced in Chapter 3 were utilized. The samples extracted by CLK_{45}, CLK_{135}, CLK_{225} , and CLK_{315} are the demultiplexed outputs in the quarter-rate architecture [18]. Figures 4.16 and 4.17 depict the simulation results of the quarter-rate and the octant-rate PDs. The amount of jitter on the control line was significantly lower in the half-rate case compared to the quarter-rate and the octant-rate PDs. The same result was achieved in the previous chapter. From lock time perspective, the octant-rate PD showed the worst case. Table 8 includes the information for bang-bang phase detection. Note that the peak-to-peak jitter values were obtained from the eye diagram measurement tool, provided in Cadence Virtuoso environment. The amount of jitter on the recovered clock was measured and reported. An example of eye diagram extraction will be shown in the transistor-level design section.

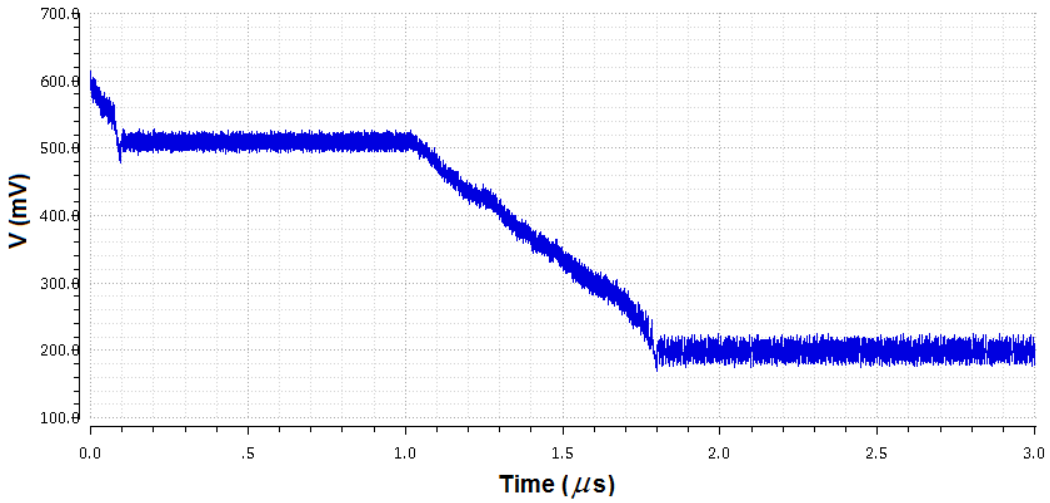


Figure 4.15: VCO control voltage for half-rate PD

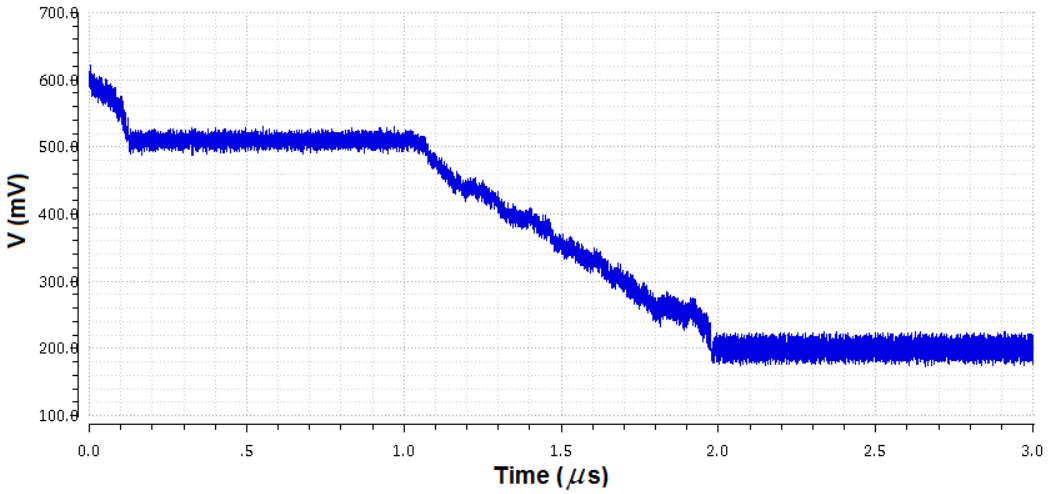


Figure 4.16: VCO control voltage for quarter-rate PD

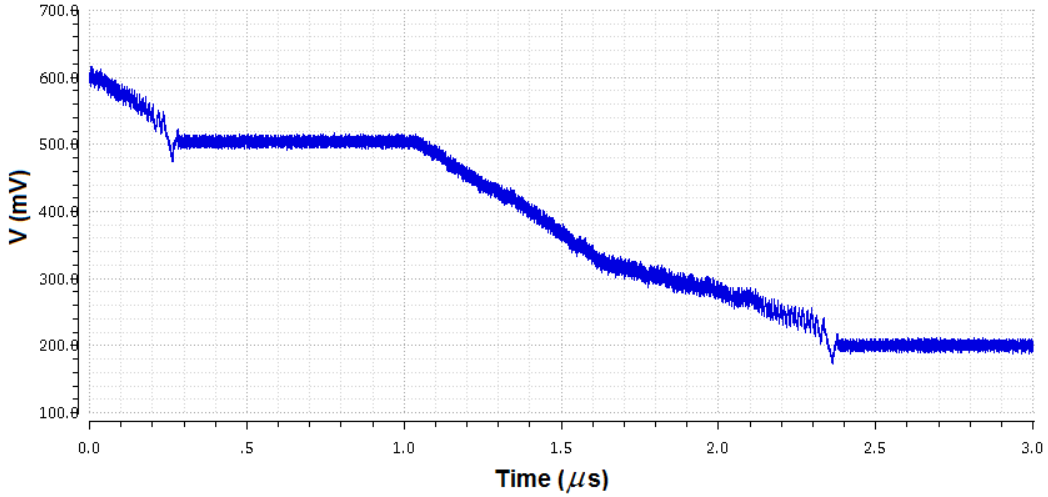


Figure 4.17: VCO control voltage for octant-rate PD

Table 8. Simulation results for various phase detectors

PD	PD Type	Timing Metrics			PRBS7 10Gbps Lock Time [ns]	PRBS7 9.5Gbps Lock Time [ns]	P-P Jitter 10.2Gbps [UI]	P-P Jitter 9.9Gbps [UI]
		Tsu [ps]	Tho [ps]	Tcq [ps]				
Half-Rate	Binary	10	10	20	≈80	≈800	≈0.065	≈0.087
Quarter-Rate	Binary	10	10	20	≈100	≈990	≈0.091	≈0.122
Octant-Rate	Binary	10	10	20	≈280	≈1380	≈0.115	≈0.134

4.2. CDR Transistor-Level Design

In this section, a CDR circuit was implemented in 45 nm technology to validate the Simulink and the Verilog-A models and outcomes. Each block was designed and simulated. Finally, the results were extracted and compared to the previous ones.

4.2.1. Sense Amplifier Flip-Flop (SAFF)

Figure 4.18 demonstrates a sense-amplifier flip-flop block diagram, which consists of two distinct parts: pulse generator (PG) and SR-latch. PG produces pulses as a result of the input data and clock transitions. Figure 4.19 shows the PG schematic. With low clock levels, the outputs of PG are precharged through M1 and M4, turning on M5 and M6 and turning off M2 and M3. Complementary inputs are applied to M7 and M8; hence, one of these transistors will be on, precharging the drain of M9 to $V_{DD} - V_{thn}$. Because M9 is off, there would be no path for the latch output to get discharged. When the clock becomes high, M1 and M4 turn off. However, M5 and M6 are still on, and depending on the situation of M7 and M8 (i.e., either M7 or M8 is on), one of the outputs will be discharged. If D becomes high, then $\bar{S}$ will be discharged, turning off M6 and turning on M3. $\bar{S}$ starts getting charged through M3. If $\bar{D}$ becomes on, $\bar{R}$ gets discharged, and therefore, M5 and M2 are turned off and on respectively. Transistor M10 was employed to prevent the system malfunction due to leakage currents. Yet, the size of the transistor could affect the delay of this stage significantly. As a result, minimum size was recommended for the transistor [23].

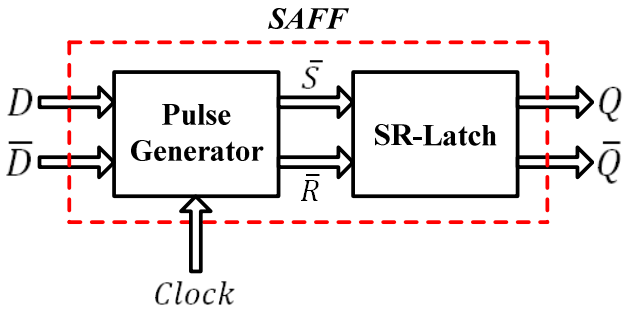


Figure 4.18: Sense-amplifier flip-flop block diagram

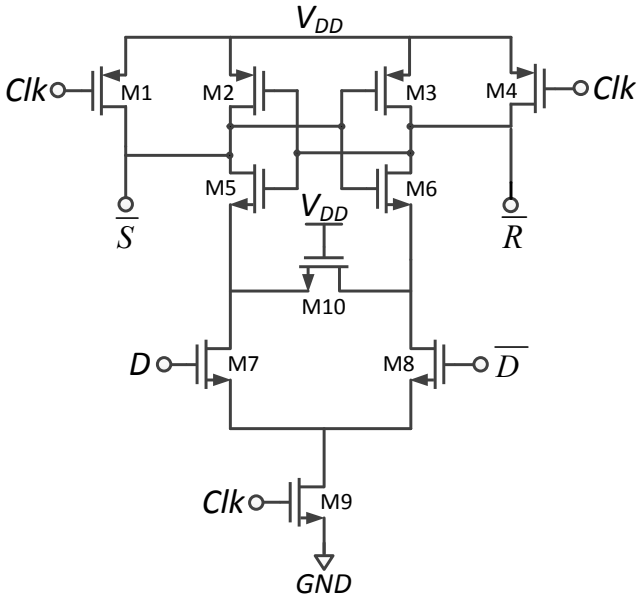


Figure 4.19: PG schematic

Figure 4.20 depicts the SR-latch schematic, demonstrated in [23]. In the proposed circuit, the contemporary SR-latch architecture was modified to overcome the nonsymmetrical problem. In the old architecture, the outputs of the latch were obtained by utilizing four large PMOS and four large NMOS transistors. However, in the new architecture, complementary CMOS design was considered. This means by using the

following equations, two large PMOS, two large NMOS, four small NMOS, and four small PMOS transistors were required to construct the new design.

$$Q = \overline{\overline{S}.(\overline{\overline{R}.Q})} = S + \overline{R}.Q \tag{4.1}$$

$$\overline{Q} = \overline{\overline{R}.(\overline{\overline{S}.\overline{Q}})} = R + \overline{S}.\overline{Q} \tag{4.2}$$

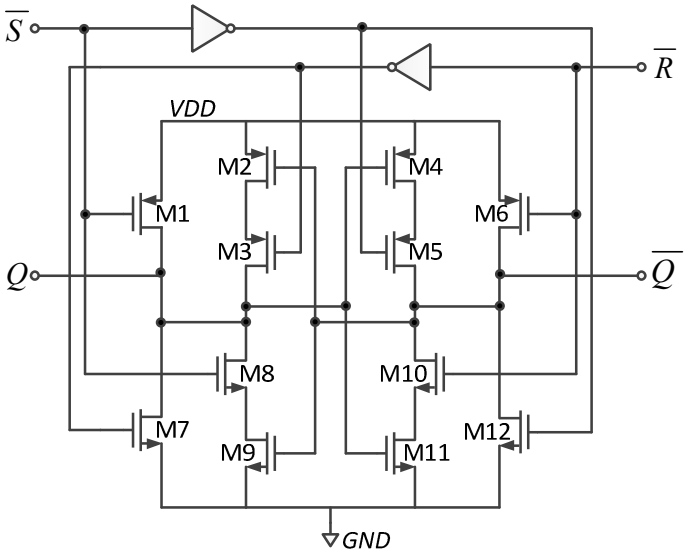


Figure 4.20: SR-latch schematic

By considering complementary CMOS design and Equation 4.1, $\overline{R}$ and Q were placed in series, and S would be in parallel with them, constructing the pull-down network. The pull-up network was constructed by a series of S and Q plus $\overline{R}$ in parallel with them. The same procedure was applied for Equation 4.2. Therefore, the architecture shown in Figure 4.20 is the combination of the previous discussions.

Figure 4.21 shows the simulation result for the designed SAFF. The first, second, and third traces depict the SAFF output, the clock, and the random data stream

respectively. As a reminder, the design was performed in 45 nm technology (i.e., length of all transistors was set to 45 nm). Moreover, the width of all transistors was minimized to diminish parasitic capacitances and time constants for fast charge and discharge purposes. The transistor sizes are provided in section B of Appendix.

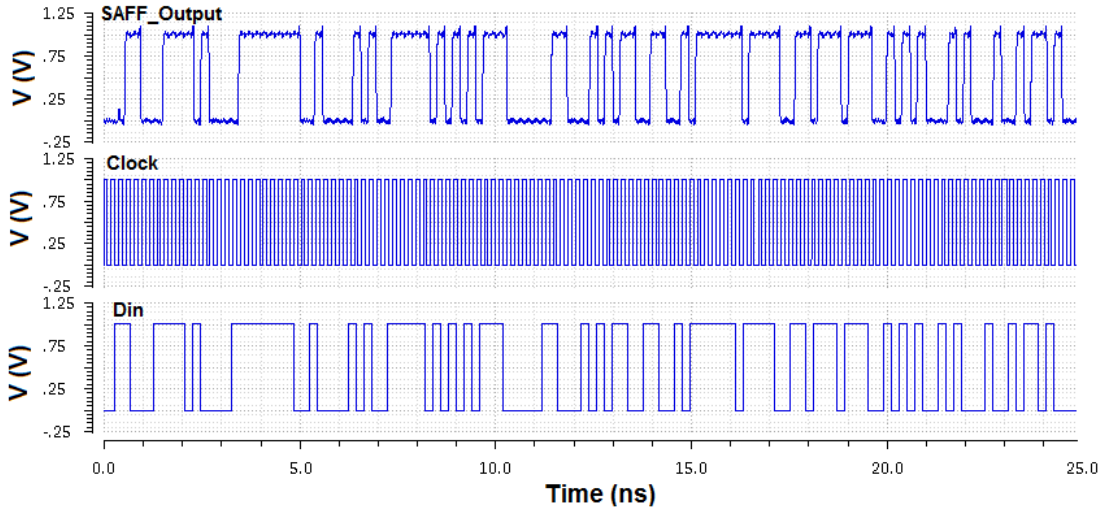


Figure 4.21: SAFF simulation result

4.2.2. Exclusive-OR

As mentioned before, an Alexander PD incorporates four DFFs, two XORs, and an inverter. Figure 4.22 depicts a complementary CMOS XOR, implemented in 45 nm technology. The circuit was designed by considering the following equation:

$$F = A \oplus B = A\overline{B} + \overline{A}B \tag{4.3}$$

To size the complementary CMOS XOR components, First, the optimum ratio of PMOS-to-NMOS width for minimum delay had to be determined. To get the ratio, an inverter

was implemented in Cadence Virtuoso. The width of NMOS was set to 120 nm, and the width of PMOS was swept from 120 nm to 240 nm. The goal was to find the crossing point of three distinct traces: low-to-high transition delay time (t_{pLH}), high-to-low transition delay time (t_{pHL}), and propagation delay time (t_p). The simulation result showed that the optimum ratio was equal to 1.45. With 120 nm for the NMOS width, the PMOS width was set to 174 nm. The procedure of finding the ratio is provided in section C of Appendix.

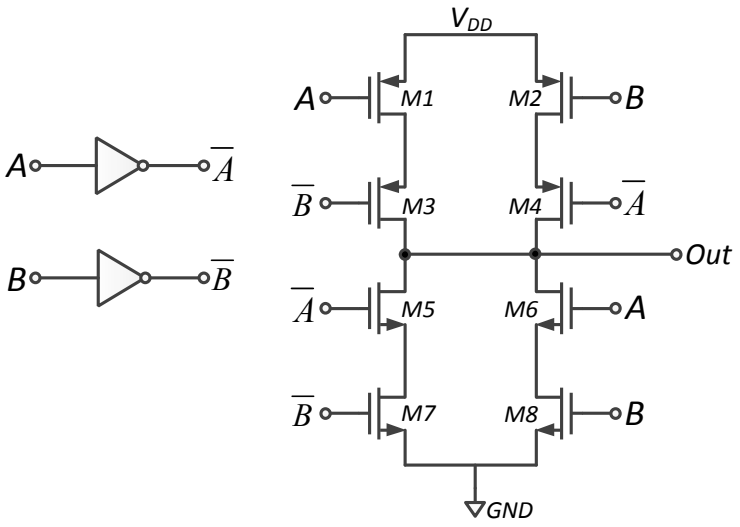


Figure 4.22: Complementary CMOS XOR schematic

The worst case for the pull-up network (PUN) consists of two PMOS transistors. Hence, the width of each PMOS transistor in PUN was set to 348 nm. The worst case for pull-down network (PDN) includes two NMOS transistors. The width of each transistor was equal to 240 nm. Figure 4.23 shows the simulation result for two periodic input pulses applied to A and B terminals. The period of each input signal was 2ns, and signal

"A" was delayed for 500 ps to show clear XOR function. Small rising and falling times were observed at the XOR output because of small transistor capacitances [1].

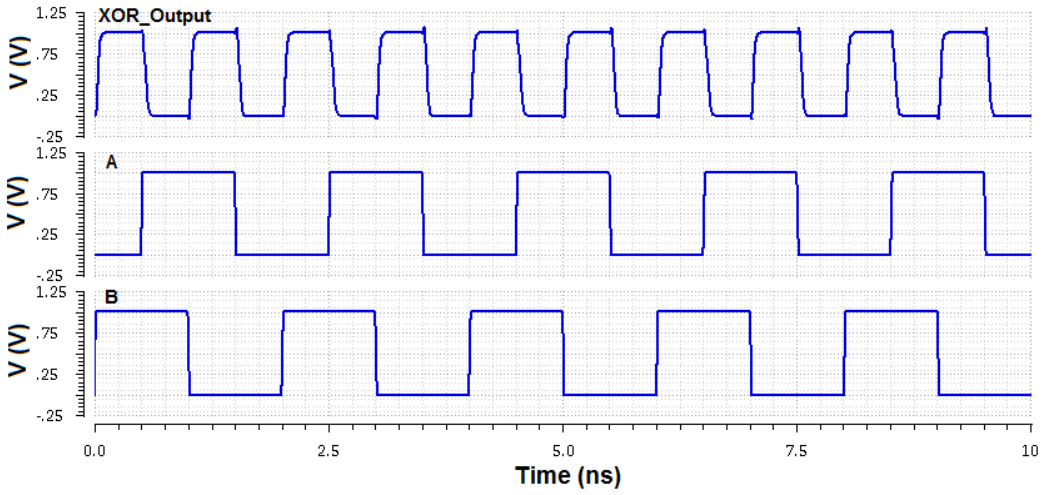


Figure 4.23: Designed XOR simulation result

4.2.3. Pseudo Random Bit Sequence (PRBS) Generator

PRBS7, which generates a data stream for a CDR, incorporates seven DFFs and an XOR. The PRBS block diagram is depicted in Figure 4.24. The DFFs were placed back-to-back, and the outputs of fourth and seventh DFFs were XORed and fed back to the first DFF as input. The designed SAFF and the complementary CMOS XOR were employed in the data generator architecture. The goal was to design a PRBS generator that could accommodate with the generated clock speed. Figure 4.25 shows the simulation result for the designed PRBS7.

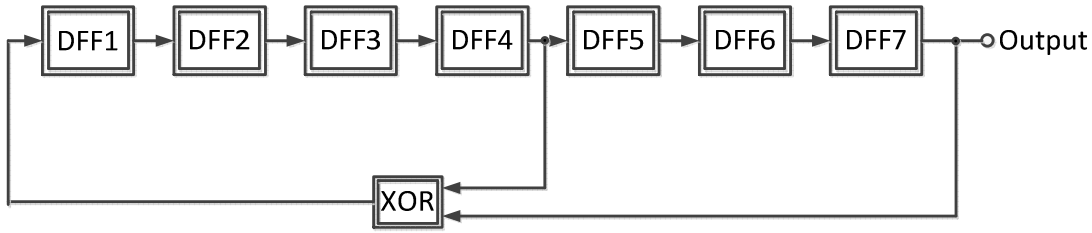


Figure 4.24: PRBS7 generator block diagram

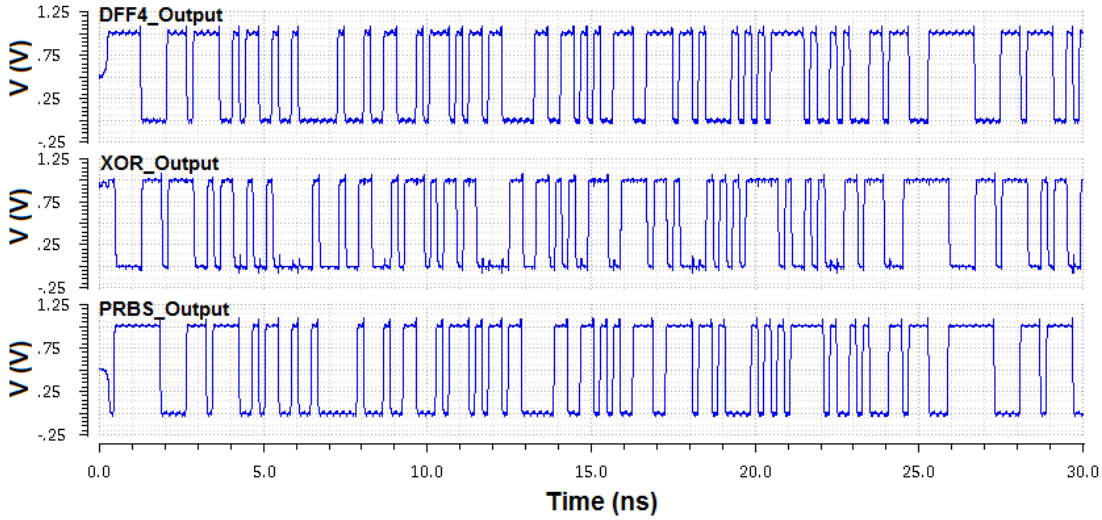


Figure 4.25: Designed PRBS generator output

4.2.4. Charge Pump

A charge pump circuit ideally incorporates two switches and two current sources to charge and discharge the loop filter [2]. PMOS and NMOS transistors play the switching role. MOSFET transistors operating in the saturation region are considered current sources. Hence, PMOS and NMOS transistors operating in the active region were provided in the CP architecture. Figure 4.26 depicts the CP schematic. High levels of the up signal at the output of the phase detector were inverted and eventually fed to the

PMOS to inject a DC current to the next stage (loop filter). To resolve the delay difference between the up and the down signals, a transmission gate was employed in the path of the down signal. The charge and discharge currents were equal to 36.97 μA. To examine the designed CP, two periodic input signals were applied to the CP. The transmission gate, the inverter, and the CP outputs are shown in Figure 4.27 respectively. The CP output clearly manifests the charge and discharge of the output capacitance. A single ended architecture was utilized in the current CDR design. Differential charge pumps and differential loop filters are usually used to reduce the amount of jitter on the VCO control line. However, power consumption and bandwidth issues prevent designers from utilizing differential circuits. Note that the bandwidth factor has to be pointed out carefully to cancel InterSymbol Interference (ISI) and noise effects.

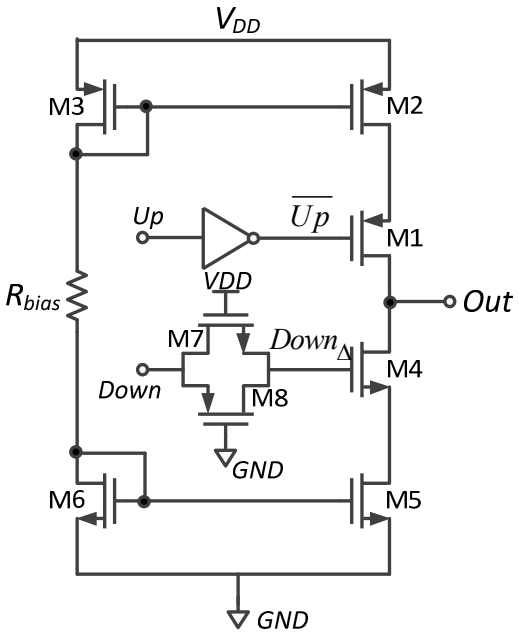


Figure 4.26: Charge pump schematic

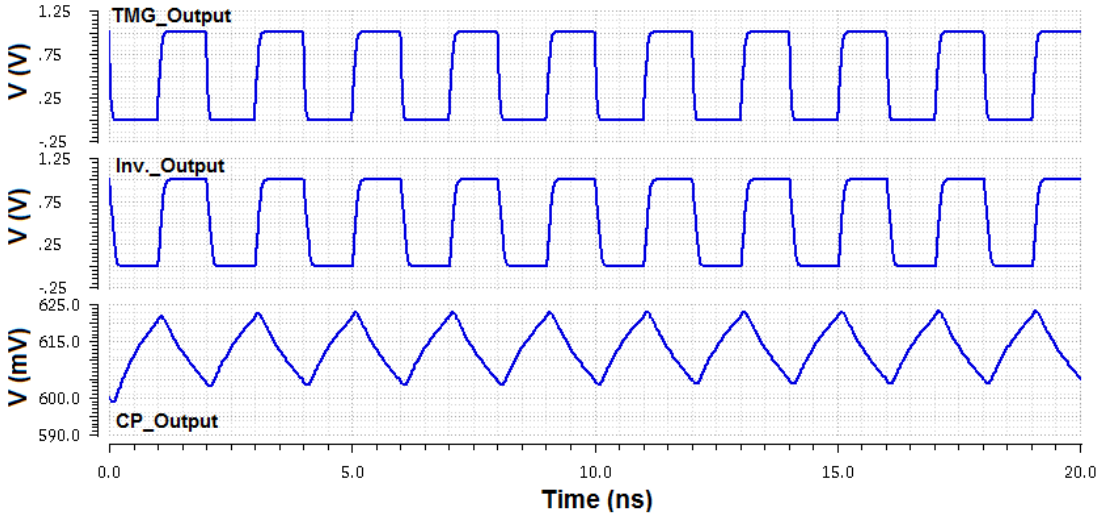


Figure 4.27: CP input and output waveforms

4.2.5. Three-Stage Ring Oscillator

The last part in the CDR circuit is the voltage-controlled oscillator (VCO), which generates the clock and varies its frequency according to the early and the late signals provided by the phase detector. For the proposed design, a three-stage ring oscillator was utilized. Figure 4.28 shows each stage of the implemented VCO using the interpolation technique for frequency tuning [2]. Two different paths were provided to vary the frequency of the generated clock. The fast path was implemented by a differential pair (M1 and M2), and the slow path consists of two differential pairs (M3,4 and M7,8). The clock frequency is adjusted based on the situation of the paths. When the fast path is on, and the slow path is off, the clock frequency is increased until it reaches the maximum oscillation frequency. However, when the fast path is off, and the slow path is on, the clock frequency is decreased to reach the minimum oscillation frequency. When both

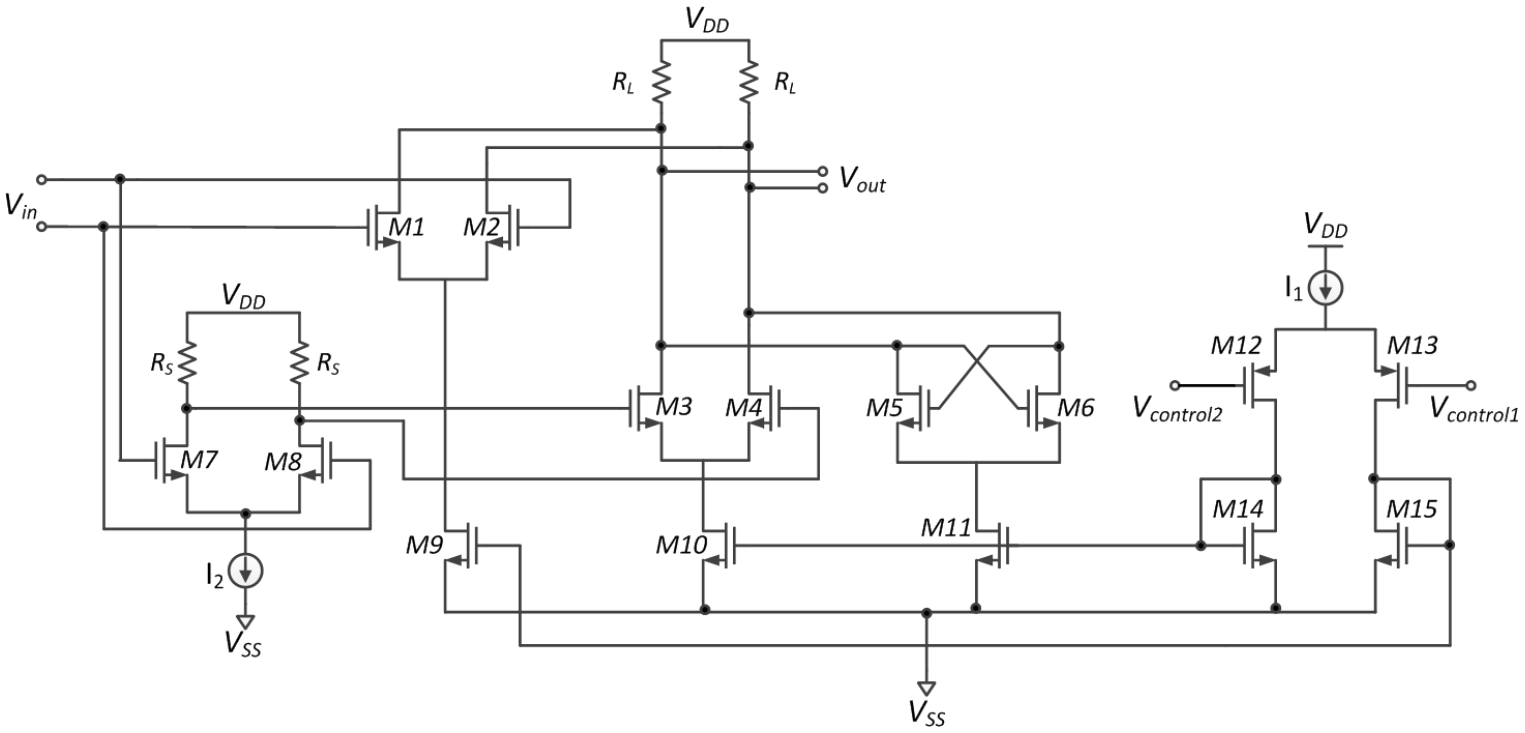


Figure 4.28: Transistor-level implementation of the VCO's each stage

paths are partially on, the clock frequency lies between the maximum and the minimum oscillation frequencies according to the input control voltages. The currents provided by the paths are summed at the output nodes and flow through the load resistors (folded cascode). Ring oscillators compared to LC counterparts provide higher tuning range, which allows the designers to select the clock frequency from a wide range of frequencies. However, LC oscillators generate clock with lower jitter compared to the ring counterparts. To reduce the amount of jitter on the generated clock in ring oscillators, usually the control voltage is provided by two distinct voltages, called "fine" and "coarse." In these cases, differential charge pumps and loop filters are utilized. The tuning range of the oscillator is adjusted by a cross-coupled transistor pair (M5 and M6), which provides a negative resistance ($-2/g_m$).

When the slow path is active, the parallel combination of the cross-coupled pair and the load resistors reduces the oscillation frequency. In a particular case in which each control voltage made a transition from one extreme (V_{DD} or V_{SS}) to another (V_{SS} or V_{DD}) in half a cycle, the clock frequency varied from 3.5 GHz to 13 GHz and vice versa. The tuning range of the oscillator was about 9.5 GHz, and K_{VCO} was approximately equal to 1.73 GHz/V.

The biasing currents for the slow and the fast paths are provided by a folding current network (M9-13 and I_1). The folding topology was utilized to avoid extra voltage headroom. The supply voltage was 1 V, and the length of all transistors was 45 nm. When the fast path is on, the tail current provided by the current mirror architecture flows through the right branch (M13 and M15) and eventually is copied to M9 according to the

ratio of the width of transistors. When the slow path is off, no current flows through the left branch (M12 and M14). When the slow path is on, and the fast path is off, the tail current steers to the left branch and is copied to M10 and M11. In partially on cases for the slow and fast paths, a fraction of the tail current flows through one of the branches, and the remaining current steers to the other branch, and eventually the clock frequency is adjusted based on the amount of the current summed at the output nodes. The biasing current for the first differential pair in the slow path (I_2) was also provided by a current mirror architecture. The small-signal gain of the VCO at 5 GHz was 17 dB, and the phase shift was -66 deg. Figure 4.29 depicts the generated clock by the three-stage ring oscillator. Two pulses in opposite directions were applied to the input control voltages with a period of 5 ns. The generated clock frequency varied from 5 GHz to 8 GHz for this particular case. Figure 4.30 depicts the clock spectrum. Phase noise at an offset frequency of 1 MHz was approximately equal to -105 dBc/Hz. The power dissipation of the ring oscillator was about 3.97 mW.

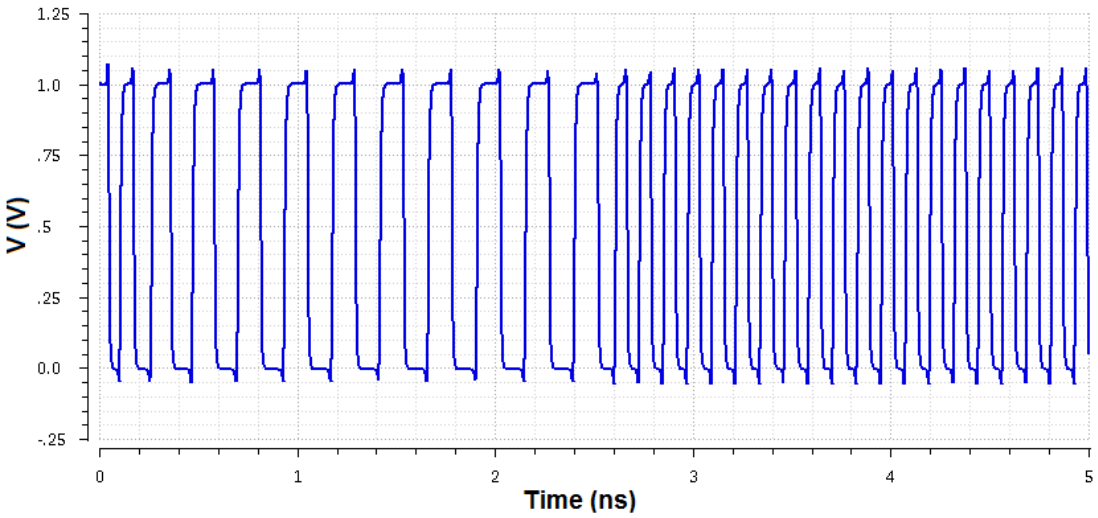


Figure 4.29: Generated clock by the ring oscillator

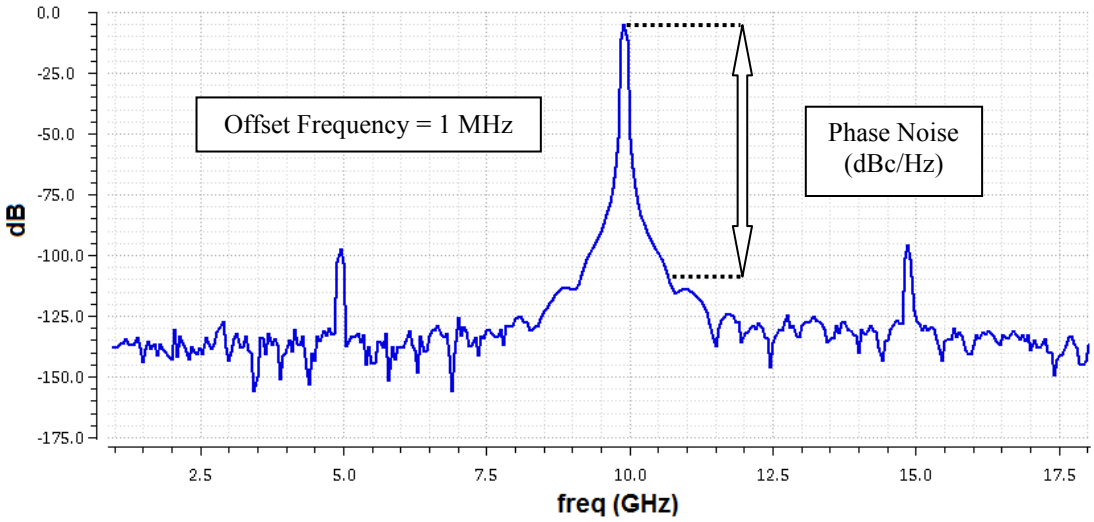


Figure 4.30: Clock spectrum

4.2.6. CDR Simulation Results

Figure 4.31 shows the control voltage of the implemented CDR. For 450 ns, a pseudo random data generator with 10 Gbps data rate was applied to the CDR. At 450 ns, the data rate was shifted to 9.7 Gbps. The CDR got locked at approximately 125 ns for the first data stream generator. After the data rate was varied, it took about 350 ns for the CDR to become locked. The free-running frequency was 9.135 GHz, and as mentioned in the previous section, K_{vco} was approximately equal to 1.73 GHz/V. The initial condition for the capacitors in the loop filter was set to 600 mV. The results manifested 0.058 UI jitter on the falling edge and 0.0469 UI on the rising edge of the generated clock. For 9.7 Gbps data generator, the peak-to-peak jitter was 0.0564 UI for the falling edge and 0.048UI for the rising edge of the clock. Figure 4.32 illustrates the CDR output eye diagram for both cases. Setup, hold, high-level sampling C2Q, and low-level sampling C2Q times were -9.6 ps, 31.15 ps, 63.98 ps, and 105.6 ps respectively.

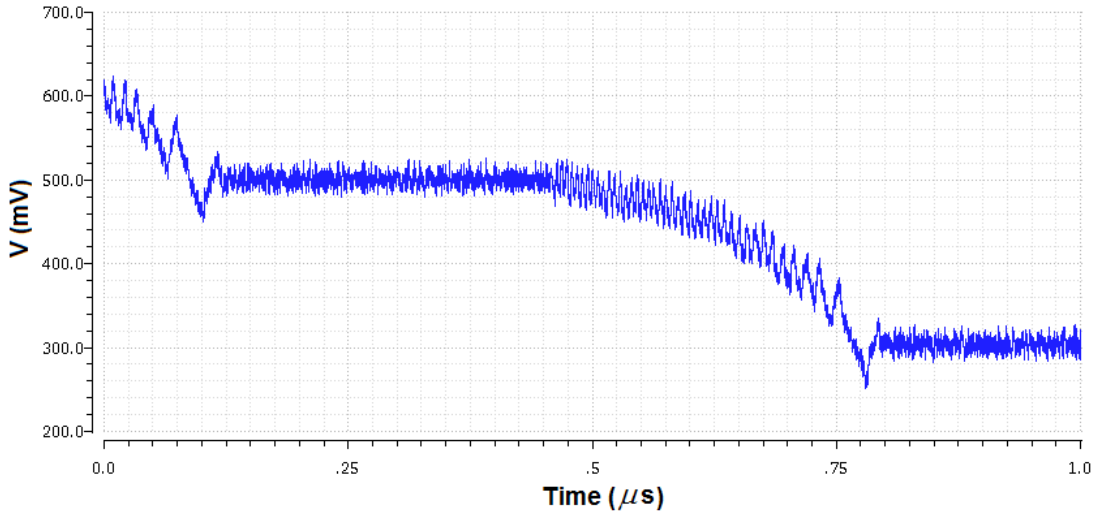


Figure 4.31: VCO control voltage

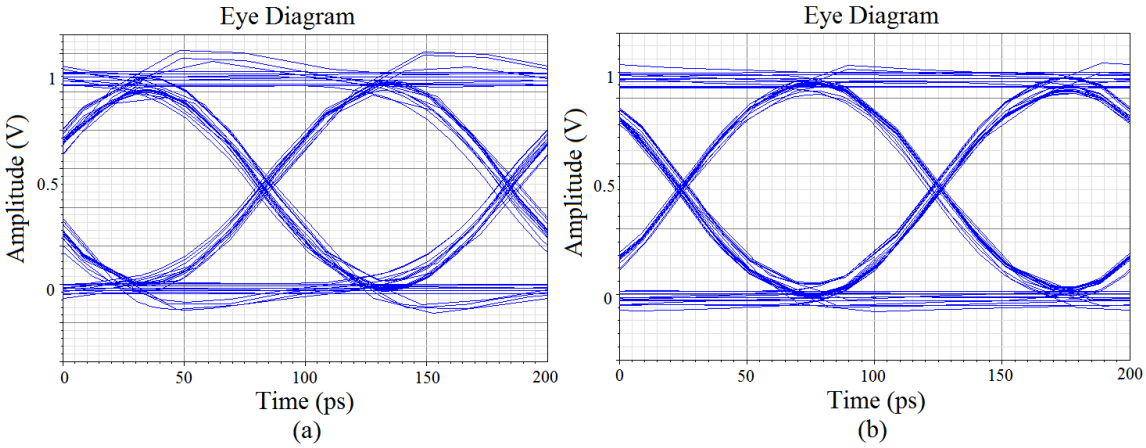


Figure 4.32: The clock eye diagram for (a) 10 Gbps data rate (b) 9.7 Gbps data rate

4.2.7. Metastability and Bang-Bang Characteristics

In this section, bang-bang characteristics were plotted for the Alexander PD, using the designed SAFF. Figures 4.33 and 4.34 compare two different cases from the timing parameter aspect. In Figure 4.33, the SAFF was designed for setup and hold

summation of 20 ps. No dead zone and metastability factors were resulted. However, in the second figure with setup and hold summation of 25 ps, the characteristic showed some nonlinearities in the rising region. The DFF calibration technique could suppress the nonlinearities by adjusting setup and hold times.

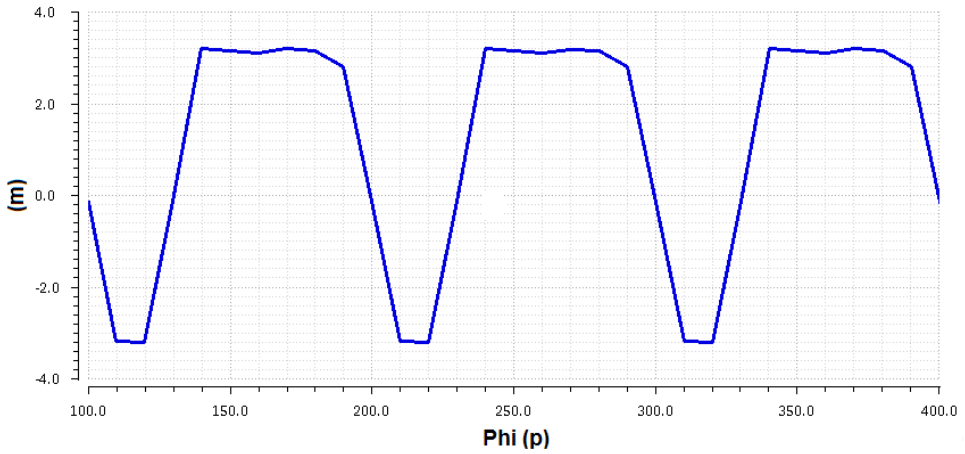


Figure 4.33: Bang-bang characteristic for $T_{su} + T_{ho} = 20$ ps

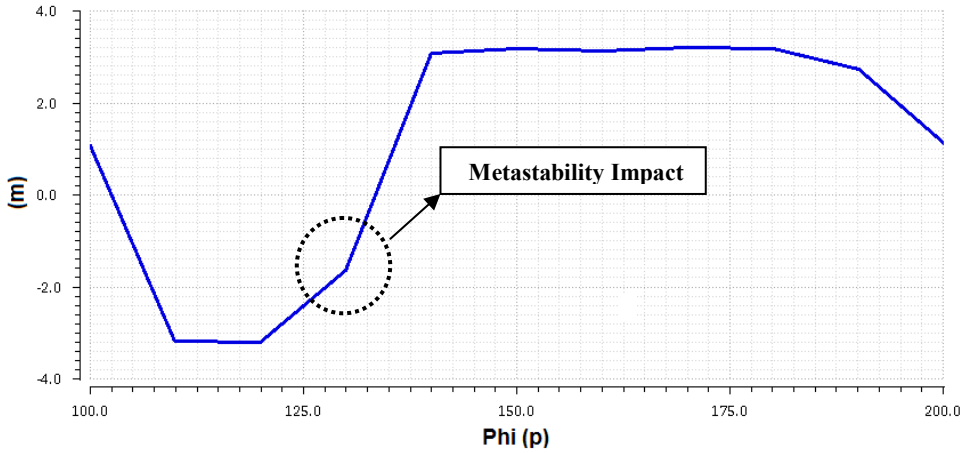


Figure 4.34: Bang-bang characteristic for $T_{su} + T_{ho} = 25$ ps

Chapter 5. Conclusions

Delay flip-flop was modeled by using Matlab® and Simulink® software. Setup, hold, and C2Q times affected the modeled DFF performance. By utilizing the implemented DFF, an Alexander phase detector was built. A CDR was constructed by means of the implemented Alexander phase detector and other elements. Moreover, realistic characteristics were generated based on various timing parameter values. CDR with the Alexander PD obtained the best result when equal values were allocated to setup and hold times. The more ideal characteristics would be obtained by allocating lower setup, hold, and C2Q times. Different PDs were modeled and simulated. By increasing the number of clock edges, the high clock frequency issue was relieved. However, the lock time and the amount of jitter on the VCO control voltage were increased significantly. To verify the results achieved by the Simulink models, the CDR was implemented in Verilog-A. The best result was achieved with equal and lower setup and hold times, which confirms accuracy of the results. The PDs with different frequency rates were implemented by Verilog-A codes and simulated. The results were almost similar to the ones in Simulink environment. Eventually, a transistor-level CDR was designed in 45nm technology. The extracted setup and hold times indicated that a time regulator has to be provided in clock and data recovery processes to push the timing metrics toward an ideal case (i.e., lower setup and hold times). Therefore, A DFF calibration technique was modeled in Matlab and Simulink, and the simulation results showed how this methodology could improve the CDR performance.

REFERENCES

[1] Rabaey, J. M.; , "Digital Integrated Circuits a Design Perspective," Prentice Hall, Second Edition, January 2002

[2] Razavi, B.; , "Design of Integrated Circuits for Optical Communications," Wiley, Second Edition, April 2012

[3] Savoj, J.; Razavi, B.; , "A 10-Gb/s CMOS Clock and Data Recovery Circuit with a Half-Rate Linear Phase Detector," *IEEE Journal of Solid-State Circuits*, vol.36, no.5, pp.761-768, May 2001

[4] Razavi, B.; , "Design of Integrated Circuits for Optical Communications," New York: McGraw-Hill, 2013

[5] Cisco Connected World Technology Report, (2010), [online]. Available: http://www.cisco.com/en/US/netsol/ns1120/index.html#~2010

[6] Gauthier C.; , (2011, November 2), "Overcoming 40G/100G SerDes Design and Implementation Challenges," [online]. Available: http://www.eetimes.com/ document.asp?doc_id=1279194

[7] Lee J.; Kundert, K.S.; Razavi, B.;, "Analysis and Modelling of Bang-Bang Clock and Data Recovery Circuits," *IEEE Journal of Solid-State Circuits*, vol.39, no.9, pp. 1571-1580, September 2004

[8] Razavi, B.; , "Principles of Data Conversion System Design," *IEEE Press*, pp.45-77, 1995

[9] Walker, R. C.; , "Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems," *International Solid-State Circuits Conference (ISSCC),* 2002.

[10] Galton, I.; , "Fractional-N PLLs," *International Solid-State Circuits Conference (ISSCC)*, 2010

[11] Alexander, J.D.H.; , "Clock Recovery from Random Binary Signals," *Electronics Letters*, vol.11, no.22, pp.541-542, October 1975

[12] Sonntag, J.L.; Stonick, J.; , "A Digital Clock and Data Recovery Architecture for Multi-Gigabit/s Binary Links," *IEEE Journal of Solid-State Circuits*, vol.41, no.8,

pp.1867-1875, August 2006

[13] Joram, N.; Barghouthi, A.; Knochenhauer, C.; Ellinger, F.; Scheytt, C.; , "Fully Integrated 50 Gbit/s Half-Rate Linear Phase Detector in SiGe BiCMOS," *Microwave Symposium Digest (MTT),* 2011

[14] Ramezani, M.; Andre, C.; Salama, C.A.T.; , "Analysis of a Half-Rate Bang-Bang Phase-Locked Loop," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* , vol.49, no.7, pp.505-509, July 2002

[15] Saffari, M.; Atarodi, M.; Tajalli, A.; , "A 1/4 Rate Linear Phase Detector for PLL-Based CDR Circuits," *IEEE International Symposium on Circuits and Systems, ISCAS Proceedings*, pp.3284, 21-24 May 2006

[16] Tontisirin, S.; Tielert, R.; , "Gb/s CMOS 1-4th-Rate CDR with Frequency Detector and Skew Calibration," *International Symposium on VLSI Design, Automation and Test*, pp.1-4, 26-28 April 2006

[17] Alavi, S.M.; Shoaei, O.; , "A 2.5-Gb/s Clock and Data Recovery Circuit with a 1/4-Rate Linear Phase Detector," *The17th International Conference on Microelectronics (ICM)*, pp.4, 13-15 December 2005

[18] Lee J.; Razavi, B.; , "A 40-Gb/s Clock and Data Recovery Circuit in 0.18-µm CMOS Technology," *IEEE Journal of Solid-State Circuits* , vol.38, no.12, pp.2181-2190, December 2003

[19] Song S. J.; Park S. M.; Yoo H. J.; , "A 4-Gb/s CMOS Clock and Data Recovery Circuit Using 1/8-Rate Clock Technique," *IEEE Journal of Solid-State Circuits*, vol.38, no.7, pp.1213-1219, July 2003

[20] Seo Y. S.; Lee J. W.; Kim H. J.; Yoo C.; Lee J. J.; Jeong C. S.; , "A 5-Gbit/s Clock and Data Recovery Circuit with 1/8-Rate Linear Phase Detector in 0.18 µm CMOS Technology," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol.56, no.1, pp.6-10, January 2009

[21] Sonntag J. L. and Stonick J., "A Digital Clock and Data Recovery Architecture for Multi-Gigabit/s Binary Links," *IEEE Journal of Solid-State Circuits,* vol.41, no.8, August 2006

[22] Hossain, M.; Carusone, A. C.; , "7.4 Gb/s 6.8 mW Source Synchronous Receiver in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol.46, no.6, pp.1337-1348, June 2011

[23] Nikolic, B.; Oklobdzija, V. G.; Stojanovic, V.; Wenyan J.; Chiu J. K.; Leung M. T.; , "Improved Sense-Amplifier Based Flip-Flop: Design and Measurements," *IEEE Journal of Solid-State Circuits*, vol.35, no.6, pp.876-884, June 2000

APPENDIX

A. Verilog-A Codes

A.1. Charge Pump

```
// VerilogA for CDR_V, CP, veriloga
`include "constants.vams"
`include "disciplines.vams"
module CP (Up,Dn,Icp, Vdd, Vss);
output Icp;    electrical Icp;     // current output
input  Up,Dn;  electrical Up,Dn;
inout Vss, Vdd;     electrical Vss,Vdd;
electrical rst;
parameter real icpn=1u;       // maximum sinking current
parameter real vth = 0.5;
parameter real icp=200e-6/2/3.14;
real subb, iout;
analog begin
subb = V(Up)-V(Dn);
iout = icp*subb;
I(Icp)<+ transition(iout,1e-15,1e-15);
end
endmodule
```

A.2. Ideal DFF

```
// VerilogA for CDR_V, DFF, veriloga
`include "constants.vams"
`include "disciplines.vams"
module DFF (q, d, clk,Vss,Vdd);
parameter integer dir = +1 from [-1:+1] exclude 0;
      // if dir=+1, rising clock edge triggers flip flop
      // if dir=-1, falling clock edge triggers flip flop
      // The discipline=voltage is used for data flow modeling
      // i.e.  no V,I feedback used for inputs d, clk, & output q
output q; voltage q;       // Q output
input clk; voltage clk;          // Clock input (edge triggered)
input d; voltage d;        // D input
integer state;             // analog integer used for flip flop
boolean state
inout Vss,Vdd;
```

```
analog begin
     @(cross(V(clk) - 0.5, dir))
         state = (V(d) > 0.5);
     V(q) <+ transition (( state ? 1 : 0 ), 0, 10f, 10f);
end
endmodule
```

A.3. Multiplexer

```
// VerilogA for CDR_V, Mux, veriloga
`include "constants.vams"
`include "disciplines.vams"
module Mux(Va, Vb,S, Vo);
input Va,Vb,S; electrical Va,Vb,S;
output Vo; electrical Vo;
real outv;
analog begin
if (V(S) > 0.5)
outv = V(Va);
else
outv = V(Vb);
V(Vo) <+ transition(outv,0,1f,1f);
end
endmodule
```

A.4. Slicer

```
// VerilogA for CDR_V, SLICER, veriloga
`include "constants.vams"
`include "disciplines.vams"
module SLICER(in, out,out_b);
input in; electrical in;
output out,out_b; electrical out,out_b;
parameter real vth = 0.5;
real outv, outvb;
analog begin
if (V(in) > 0.5)
 begin
   outv = 1;
   outvb = 0;
 end
else
 begin
  outv = 0;
  outvb = 1;
 end
```

```
V(out) <+ transition(outv,0,10f,10f);
V(out_b) <+ transition(outvb,0,10f,10f);
end
endmodule
```

A.5. Voltage-Controlled Oscillator

```
// VerilogA for CDR_V, VCO, veriloga
`include "constants.vams"
`include "disciplines.vams"
module VCO(Vc,Out, Vss, Vdd);
input Vc;  electrical Vc;
output Out;      electrical Out;
inout Vss, Vdd;  electrical Vss,Vdd;
parameter real f0 = 9.7e9;
parameter real Kvco = 1e9;
real f, amp, offset;
analog begin
f = f0 + Kvco*V(Vc);
amp = (V(Vdd)-V(Vss))/2;
offset = V(Vss)+amp;
V(Out) <+ amp*sin(2*`M_PI*idtmod(f,0,1))+offset;
end
endmodule
```

A.6. Exclusive-OR

```
// VerilogA for CDR_V, XOR, veriloga
`include "constants.vams"
`include "disciplines.vams"
module XOR (in_a, in_b,out);
 input in_a, in_b;
 output out;
 electrical in_a, in_b;
 electrical out;
real a,b,c;
 analog begin
if (V(in_a) > 0.5)
 a = 1;
else
 a = 0;
if (V(in_b) > 0.5)
 b = 1;
else
 b = 0;
c = a+b;
if (c > 1.5)
```

```
  c = 0;
  V(out) <+ transition(c,0,10f,10f);
 end
endmodule
```

A.7. DFF Metastability

```
// VerilogA for CDR, DFF_Metastability, veriloga
`include "constants.vams"
`include "disciplines.vams"
module DFF_Metastability(in, clk, vdd, vss, out);
input in, clk;
output out;
inout vdd, vss;
electrical in, clk, out, vdd, vss;
parameter real Tsu=19p;                 //Defining setup time
parameter real Tho=1p;                 //Defining hold time
parameter real Tc2q=40p;               //Defining C2Q time
parameter real trise=10f;                  //Rise time used at the
final stage
parameter real tfall=10f;                  //Fall time used at the
final stage
parameter integer bit_num = 8 from [2:15];
parameter integer seed = 1 from [1:inf];
real tdata_rise;                            //Data rise time
real tclk_rise;                             //Clock rise time
real tdata_fall;                            //Data fall time
real state1;
real state2;
real state3;
real state4;
real state;
real state5;
integer x, a1, a2, a3, a4, b, mask;         //Parameters for
generating random data stream
integer Y;
analog begin
        @(initial_step) begin
          case (1)
          (bit_num ==  2): begin a1=0; a2= 1; a3= 0; a4= 0; end
// 2 [0,1]
            (bit_num ==  3): begin a1=0; a2= 2; a3= 0; a4= 0; end
// 3 [0,2]              /* Generating a random pulse
            (bit_num ==  4): begin a1=0; a2= 3; a3= 0; a4= 0; end
// 4 [0,3]                  for the metastability case */
            (bit_num ==  5): begin a1=1; a2= 4; a3= 0; a4= 0; end
// 5 [1,4]
```

```
            (bit_num ==  6): begin a1=0; a2= 5; a3= 0; a4= 0; end
// 6 [0,5]
            (bit_num ==  7): begin a1=0; a2= 6; a3= 0; a4= 0; end
// 7 [0,6]
            (bit_num ==  8): begin a1=1; a2= 2; a3= 3; a4= 7; end
// 8 [1,2,3,7]
            (bit_num ==  9): begin a1=3; a2= 8; a3= 0; a4= 0; end
// 9 [3,8]
            (bit_num == 10): begin a1=2; a2= 9; a3= 0; a4= 0; end
//10 [2,9]
            (bit_num == 11): begin a1=1; a2=10; a3= 0; a4= 0; end
//11 [1,10]
            (bit_num == 12): begin a1=0; a2= 3; a3= 5; a4=11; end
//12 [0,3,5,11]
            (bit_num == 13): begin a1=0; a2= 2; a3= 3; a4=12; end
final stage
parameter integer bit_num = 8 from [2:15];
parameter integer seed = 1 from [1:inf];
real tdata_rise;                              //Data rise time
real tclk_rise;                               //Clock rise time
real tdata_fall;                              //Data fall time
real state1;
real state2;
real state3;
real state4;
real state;
real state5;
integer x, a1, a2, a3, a4, b, mask;         //Parameters for
generating random data stream
integer Y;
analog begin
        @(initial_step) begin
          case (1)
           (bit_num ==  2): begin a1=0; a2= 1; a3= 0; a4= 0; end
// 2 [0,1]
            (bit_num ==  3): begin a1=0; a2= 2; a3= 0; a4= 0; end
// 3 [0,2]              /* Generating a random pulse
            (bit_num ==  4): begin a1=0; a2= 3; a3= 0; a4= 0; end
// 4 [0,3]                 for the metastability case */
            (bit_num ==  5): begin a1=1; a2= 4; a3= 0; a4= 0; end
// 5 [1,4]
            (bit_num ==  6): begin a1=0; a2= 5; a3= 0; a4= 0; end
// 6 [0,5]
            (bit_num ==  7): begin a1=0; a2= 6; a3= 0; a4= 0; end
// 7 [0,6]
            (bit_num ==  8): begin a1=1; a2= 2; a3= 3; a4= 7; end
// 8 [1,2,3,7]
            (bit_num ==  9): begin a1=3; a2= 8; a3= 0; a4= 0; end
// 9 [3,8]
```

```
            (bit_num == 10): begin a1=2; a2= 9; a3= 0; a4= 0; end
//10 [2,9]
            (bit_num == 11): begin a1=1; a2=10; a3= 0; a4= 0; end
//11 [1,10]
            (bit_num == 12): begin a1=0; a2= 3; a3= 5; a4=11; end
//12 [0,3,5,11]
            (bit_num == 13): begin a1=0; a2= 2; a3= 3; a4=12; end
        state=(Y*state5)+(!Y*b);
//Identifying the state of the output signal
        V(out) <+ transition((state ? 1:0),2p,trise,tfall);
//DFF Metastability output
end
endmodule
```

A.8. 2-Bit Adder

```
// VerilogA for CDR, 2_bit_adder, veriloga
`include "constants.vams"
`include "disciplines.vams"
module adder(in1,in2,out);
input in1, in2;
output out;
electrical in1, in2;
electrical out;
analog begin
     I(out) <+ transition(I(in1) + I(in2),0,10f,10f);
end
endmodule
```

A.9. 4-Bit Adder

```
// VerilogA for CDR, 4_bit_adder, veriloga
`include "constants.vams"
`include "disciplines.vams"
module adder(in1,in2,in3,in4,out);
input in1,in2,in3,in4;
output out;
electrical in1,in2,in3,in4,out;
real sum;
analog begin
     sum= I(in1)+I(in2)+I(in3)+I(in4);
     I(out) <+ transition(sum,0,10f,10f);
end
endmodule
```

A.10. 8-Bit Adder

```
// VerilogA for CDR, 8_bit_Adder, veriloga
`include "constants.vams"
`include "disciplines.vams"
module Adder(in1,in2,in3,in4,in5,in6,in7,in8,out);
input in1,in2,in3,in4,in5,in6,in7,in8;
output out;
electrical in1,in2,in3,in4,in5,in6,in7,in8,out;
real sum;
analog begin
      sum=
I(in1)+I(in2)+I(in3)+I(in4)+I(in5)+I(in6)+I(in7)+I(in8);
      I(out) <+ transition(sum,0,10f,10f);
end
endmodule
```

A.11. Delay

```
// VerilogA for CDR, Delay, veriloga
`include "constants.vams"
`include "disciplines.vams"
module Delay(in, out);
input in;
output out;
electrical in, out;
parameter real delay=0.25*(1/10e9);
analog begin
       V(out) <+ transition(V(in), delay,10f,10f);
end
endmodule
```

B. Designed CDR Transistor Sizes (L = 45 nm)

B.1. Pulse Generator (PG)

Table B.1. Pulse generator device sizes

Transistor	Width(nm)	Transistor	Width(nm)
M1	120	M6	240
M2	120	M7	240
M3	120	M8	240
M4	120	M9	240
M5	240	M10	120

B.2. SR-latch

Table B.2. SR-latch device sizes

Transistor	Width(nm)	Transistor	Width(nm)
M1	320	M7	540
M2	320	M8	540
M3	320	M9	540
M4	320	M10	540
M5	320	M11	540
M6	320	M12	540

B.3. Exclusive-OR

Table B.3. XOR device sizes

Transistor	Width(nm)	Transistor	Width(nm)
M1	480	M7	240
M2	480	M8	240
M3	480	M9	240
M4	480	M10	240
Mpinv	5.1 µm	Mninv	3 µm

B.4. Charge Pump

Table B.4. Charge pump device sizes

Transistor	Width	Transistor	Width
M1	12.9 µm	M5	10 µm
M2	12.9 µm	M6	120 nm
M3	120 nm	M7	120 nm
M4	10 µm	M8	120 nm
Mpinv	175 nm	Mninv	120 nm
Rbias		1 KΩ	

B.5. Voltage-Controlled Oscillator

Table B.5. Voltage-controlled oscillator device sizes

Transistor	Width	Transistor	Width
M1,2	14 µm	M10	1.6 µm
M3,4	8.2 µm	M11	10 µm
M5,6	3 µm	M12,13	10 µm
M7,8	1.18 µm	M14	1 µm
M9	3 µm	M15	120 nm
RL,RS		1 KΩ	
I1		42 µA	
I2		112 µA	

C. Optimum PMOS-to-NMOS Aspect Ratio

To find the optimum PMOS-to-NMOS aspect ratio, the schematic shown in Figure C.1 was utilized. The ratio provides the minimum delay between input and output signals of the inverter. To do that, a periodic pulse with a period of 2 ns was applied to the input, and a 10 fF capacitor was placed at the output. Figure C.2 depicts the input and output signals applied to the inverter. The output signal showed inversion with limited rising and falling times. The optimum ratio was obtained when low-to-high and high-to-low propagation delays shared a common point. Figure C.3 illustrates the delay plots. T_{pLH} and T_{pHL} graphs were obtained by using "delay" command. The PMOS width was swept from120 ns to 240 ns, and NMOS width was 120 ns. Note that T_p is the average of T_{pLH} and T_{pHL}. The optimum result was achieved for 174.2 ns. It means the ratio of PMOS to NMOS was approximately equal to 1.45.

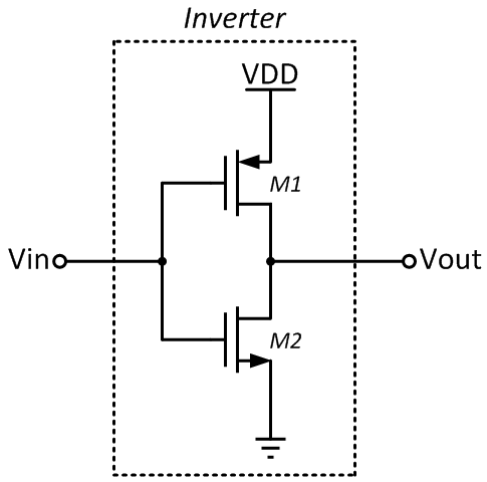


Figure C.1: Inverter schematic

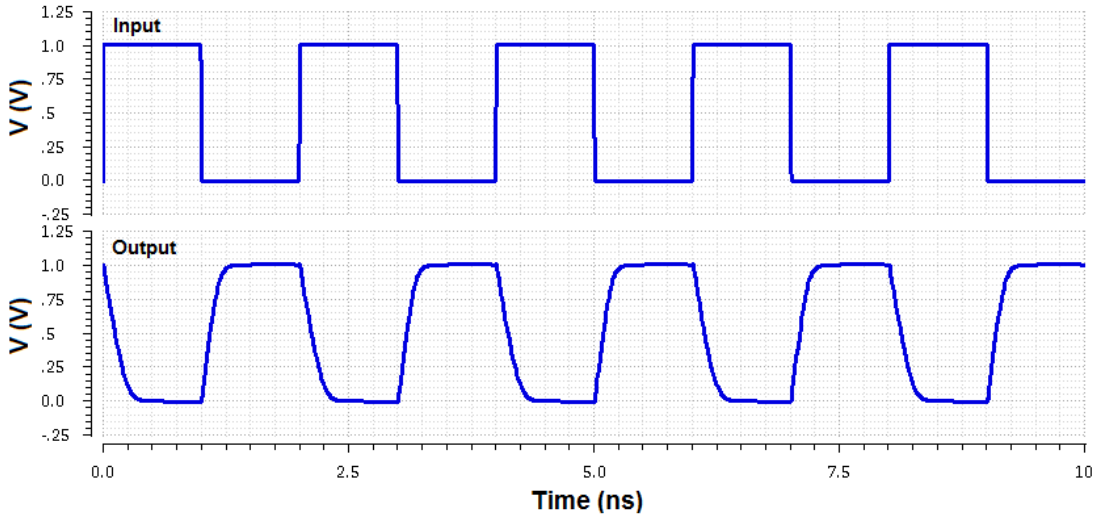


Figure C.2: Input and output signals of the inverter

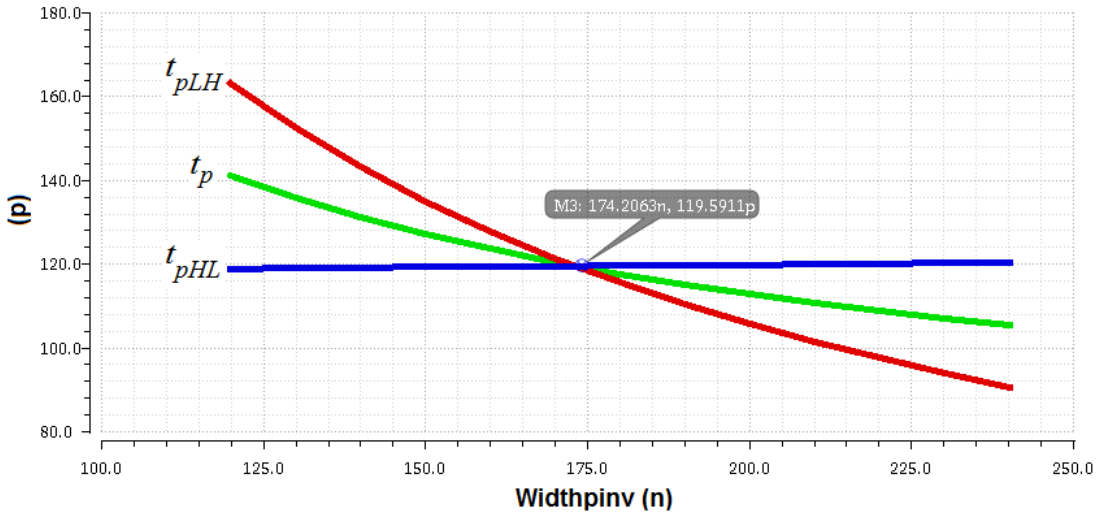


Figure C.3: Propagation delay graphs

D. Three-Stage VCO Using the Interpolation Technique

In Chapter 4, some of the designed ring oscillator simulation results were presented. In this section, the discussed results are shown graphically. Figure D.1 manifests the bode plots for each stage of the three-stage ring oscillator using the interpolation technique. At 5 GHz clock frequency, the stage amplitude showed that the gain of each stage was equal to 17 dB, which was considerably high. Moreover, the phase shift at 5 GHz was -66 deg. The VCO phase noise is shown in Figure D.2. To generate phase noise and clock spectrum figures, "pss" and "pnoise" analyses were performed. After running these analyses, the following functions located in the calculator of Cadence Virtuoso were utilized to generate the pss and phase noise graphs.

dBm (spectralPower((v("/VCO_Output" ?result "pss_fd") / 10000)
v("/VCO_Output ?result "pss_fd")))

phaseNoise (1 "pss_fd" ?result "pnoise")

clip (dB20(dft(v("/VCO_Output" ?result "tran") 0 2e-8 262144 "Hanning" 1
"default")) 1e+09 1.8e+10)

dft (v("/VCO_Output" ?result "tran") 0 5e-10 64 "Rectangular" 1 "default")

Discrete-fourier transform (DFT) command was used to convert the time domain to the frequency domain. Figure D.3 shows the DFT of the VCO output. The X-axis is in frequency domain. The DFT graph includes information about the harmonics. In this particular case, the fundamental harmonic occurred at 10 GHz.

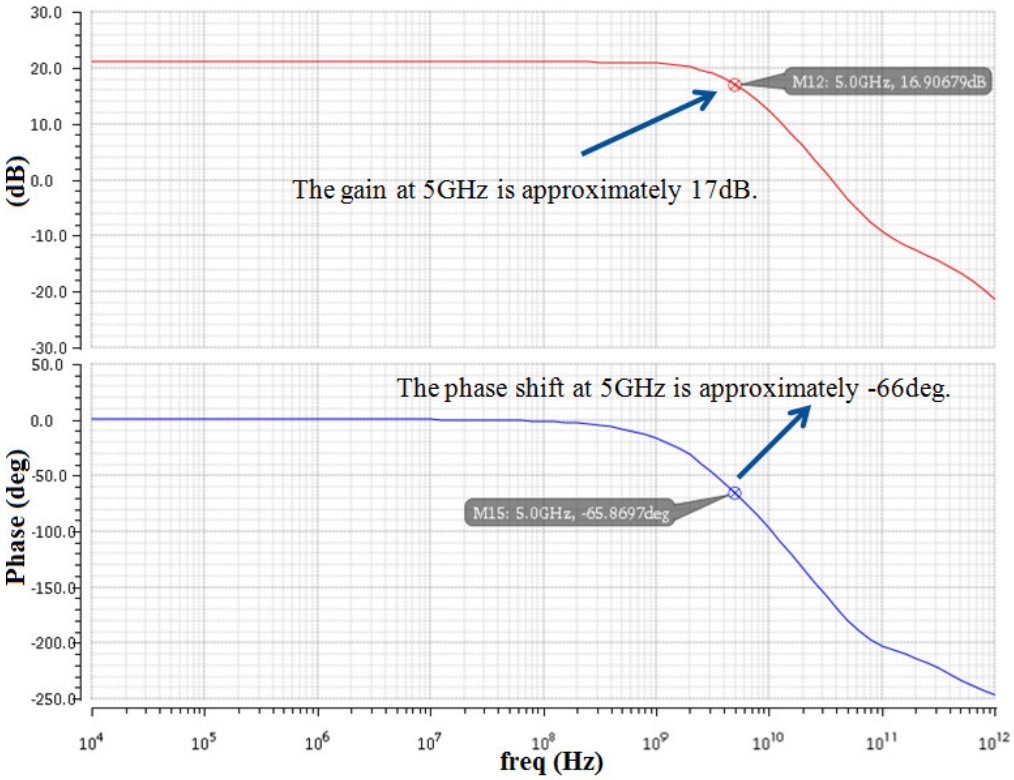


Figure D.1: Bode plot of each stage of the designed oscillator

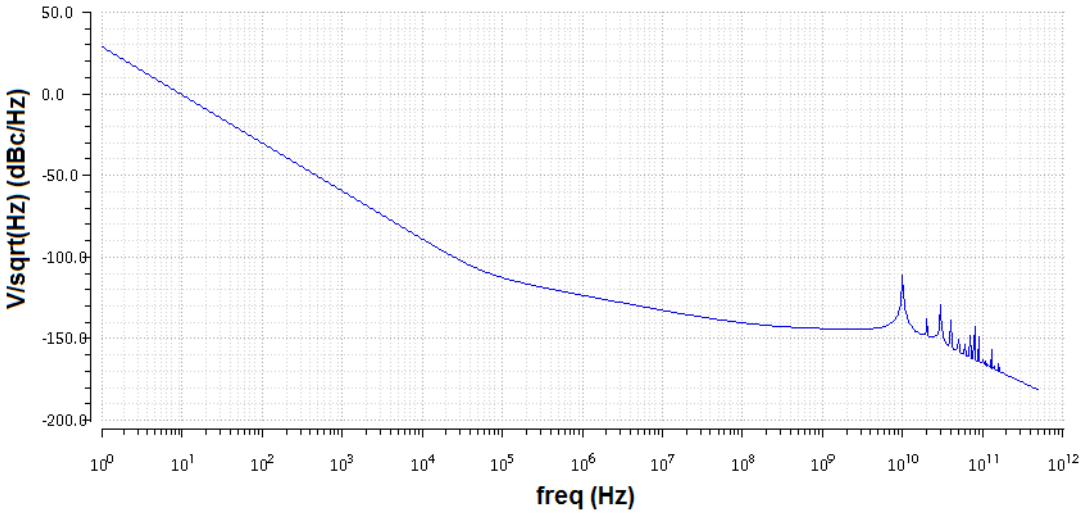


Figure D.2: Phase noise (PN)

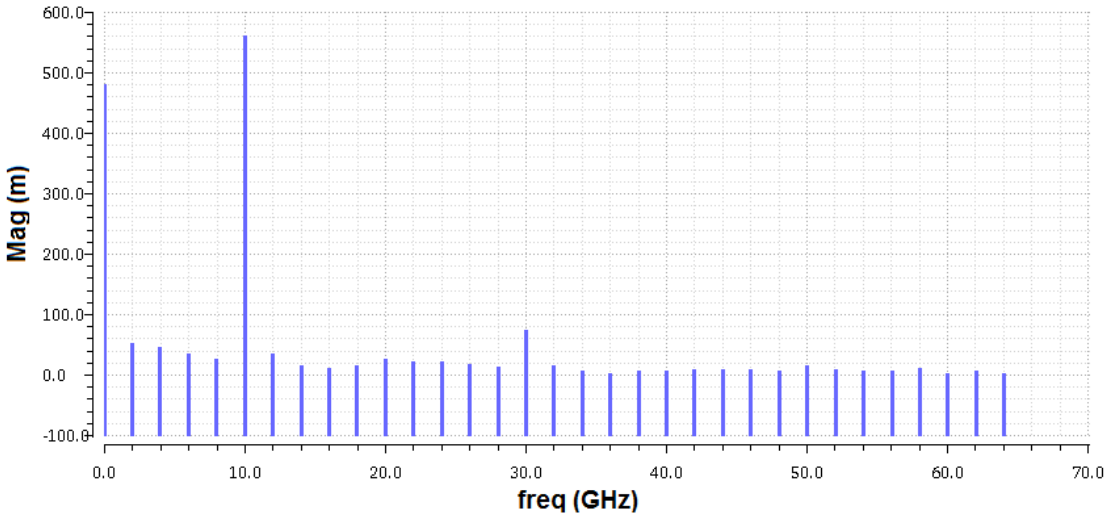
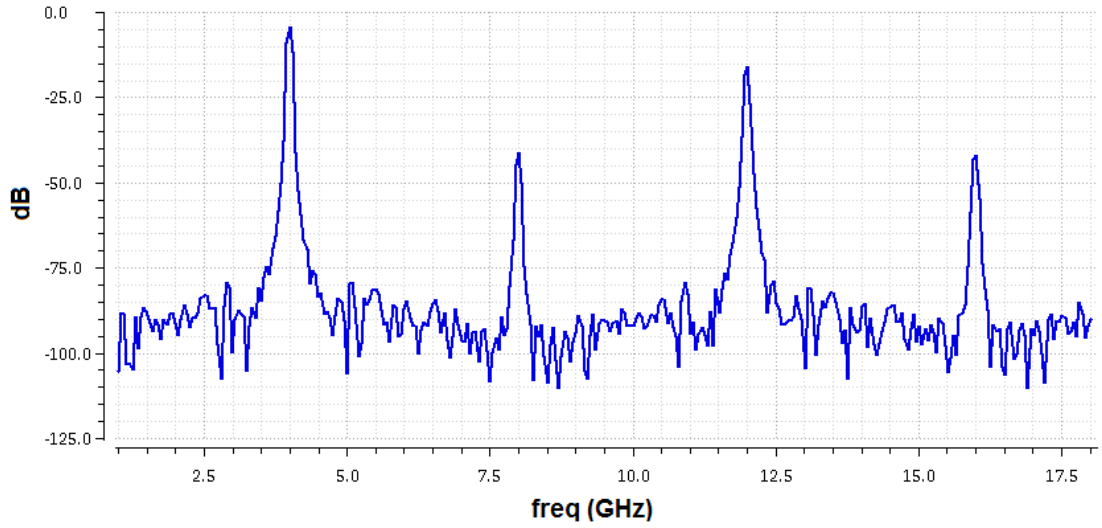
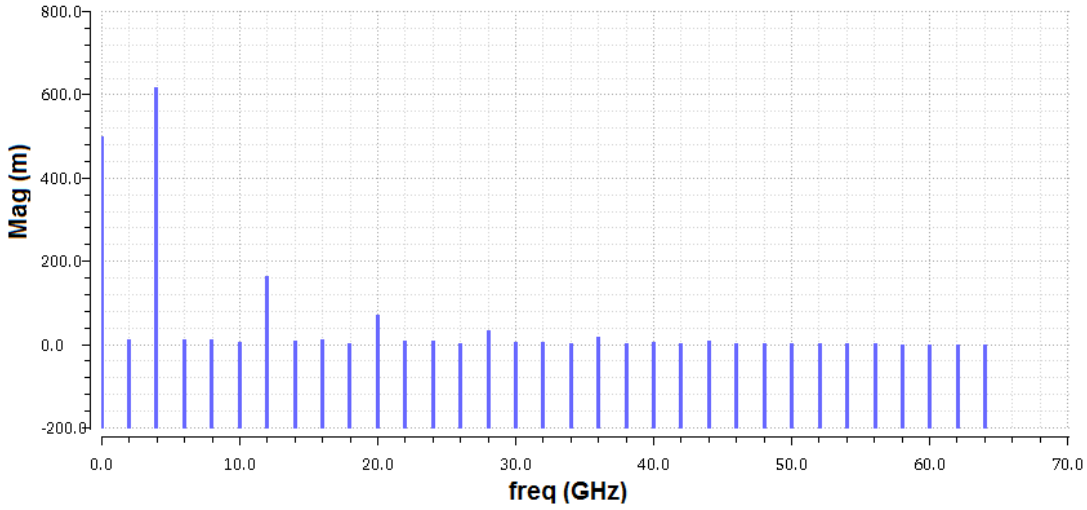


Figure D.3: DFT of the VCO output for $V_{control1}$ = 400 mV and $V_{control2}$ = 1 V

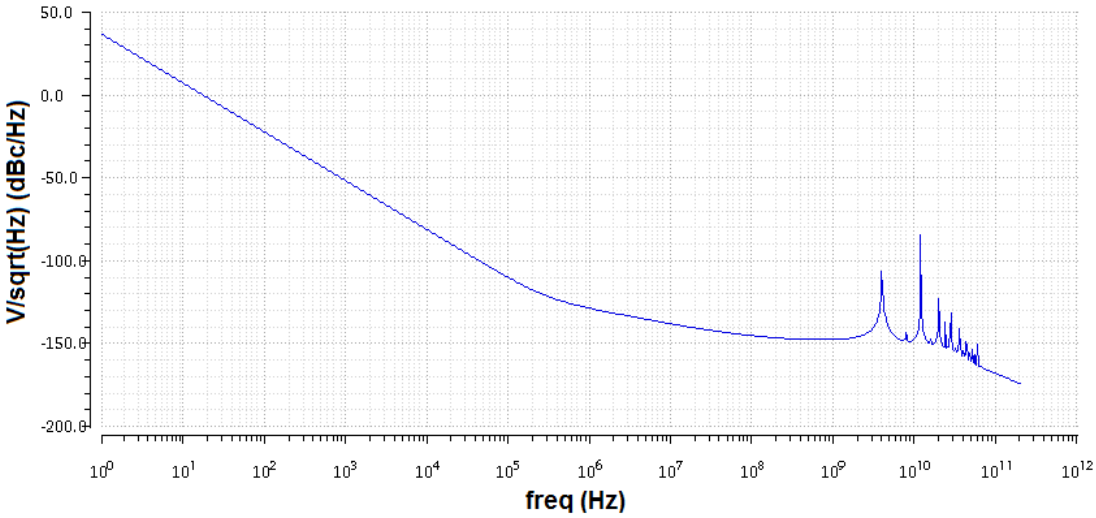
The following results demonstrate ring oscillator results for different cases.



(a)

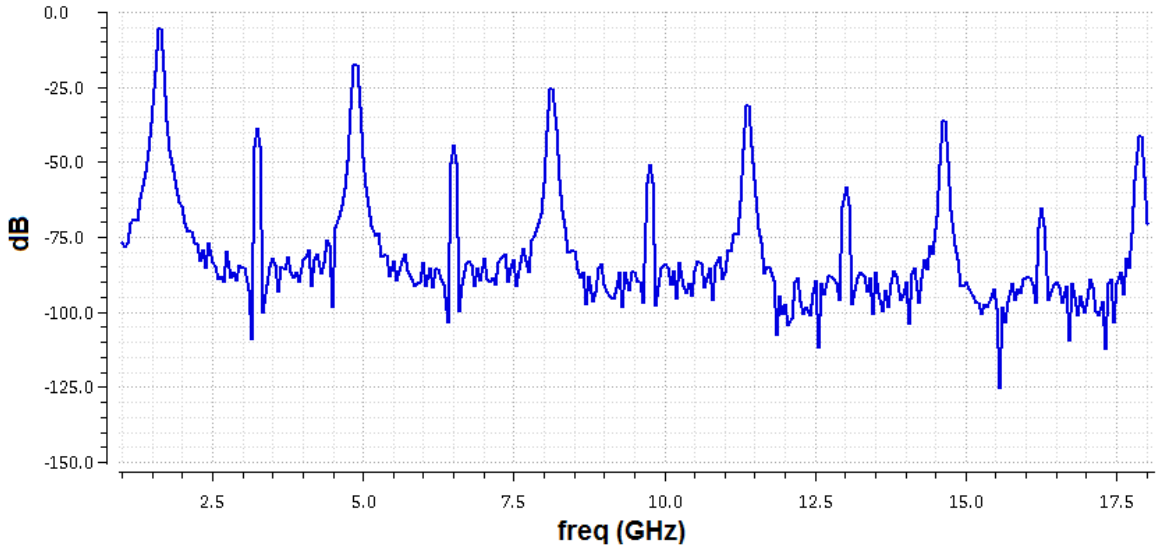


(b)

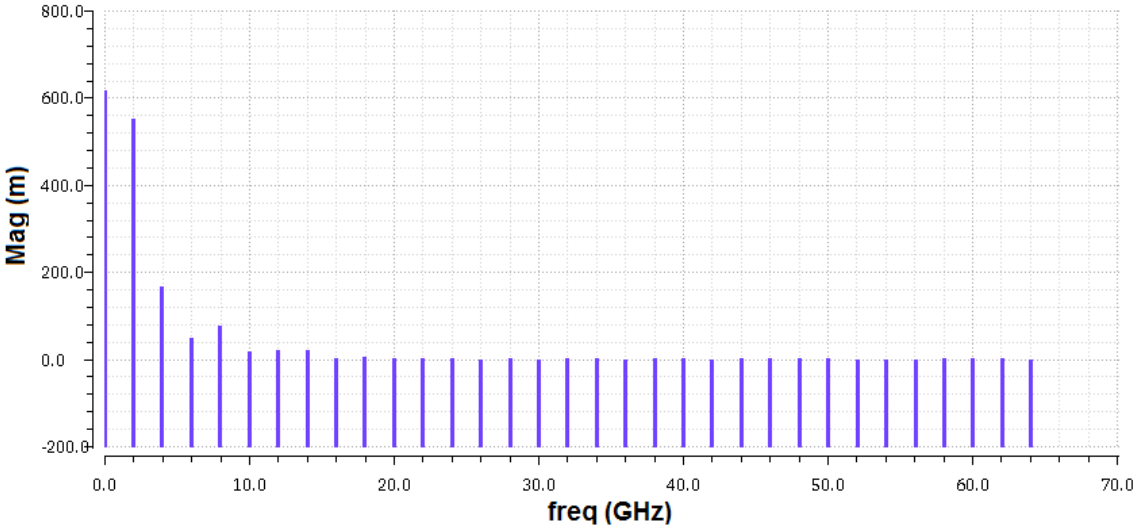


(c)

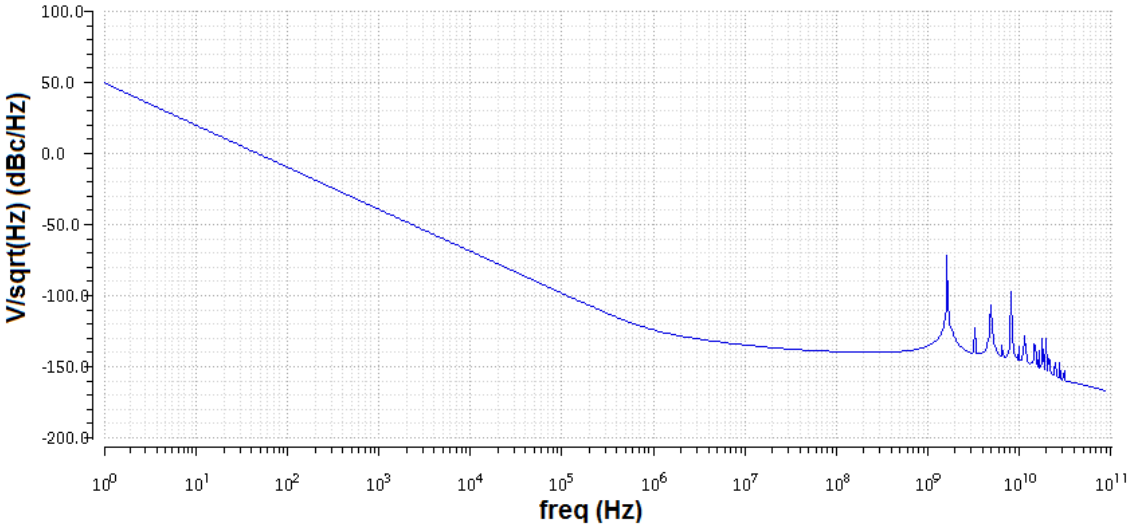
Figure D.4: $V_{control1}$ = 1 V and $V_{control2}$ = 0 V (a) clock spectrum (b) DFT (c) PN


0.0
-25.0
-50.0
-75.0
-100.0
-125.0
-150.0
dB
2.5
5.0
7.5
10.0
12.5
15.0
17.5
freq (GHz)

(a)


800.0
600.0
400.0
200.0
0.0
-200.0
Mag (m)
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
freq (GHz)

(b)



(c)

Figure D.5: $V_{control1}$ = 0 V and $V_{control2}$ = 400 mV (a) clock spectrum (b) DFT (c) PN